Sílvia Buira Clua

ONTOLOGY-BASED CONSTRUCTION OF OUTRANKING RELATIONS FOR DECISION SUPPORT

Master Thesis

Directed by Dra. Aïda Valls Mateu
Research Group: ITAKA

Master on Computer engineering: Computer Security and Intelligent Systems

UNIVERSITAT ROVIRA I VIRGILI

Tarragona 2017

Agraïments

En primer lloc, voldria expressar el meu agraïment a la meva tutora del màster, l'Aïda Valls. Primer, per proposar-me de fer aquest treball, amb el que he gaudit fent-lo. I també per la seva ajuda, per transmetre'm el seu coneixement, per les seves idees i pel seu suport.

Després voldria agrair-li al meu marit, Àngel, pel seu suport moral i ajuda logística.

I finalment, voldria dedicar-lo als meus tres fills: Àngel, Muntsa i Llibert. Espero que el temps que no els he dedicat, per poder fer aquesta tesis, els serveixi d'exemple d'esforç i que vegin que qualsevol edat és bona per continuar estudiant.

INDEX

List of Figures

List of Tables

Abbreviations

AHP	Analytic Hierarchy Process
DAMASK	Data-Mining Algorithms with Semantic Knowledge
DM	Decision Maker
ELECTRE	ELimination Et Choix Traduisant la REalité
ITAKA	Intelligent Technologies for Advanced Knowledge Acquisition
MAUT	Multi-Attribute Utility Theory
MCDA	Multi-Criteria Decision Aiding
OWA	Ordered Weighted Averaging
OWL	Ontology Web Language
RDF	Resource Description Framework
UMCDA-ML	Universal Multi Criteria Decision Analysis Modelling Language
W3C	World Wide Web Consortium

1 Introduction

People are continuously facing the problem of making correct decisions both in our professional and private lives. These decisions may be as trivial as choosing a film in Netflix or as relevant as buying a house or deciding where a road should be built.

Multi-Criteria Decision Aiding (MCDA) methods have been developed to support the Decision Maker (hereafter the DM) in his/her unique and personal decision process.

MCDA is a discipline that encompasses mathematics, management, informatics, psychology, social science and economics. Its application is even wider as it can be used to solve any problem where a significant decision needs to be made. These decisions can be either tactical or strategic, depending on the time perspective of the consequences, as shown in Table 1 [Ish13].

Decision	Time perspective	Novelty	Degree of structure	Automation
Strategic	Long term	New	Low	Low
Tactical	Medium term	Adaptive	Semi-structured	Middle
Operational	Short term	Every day	Well defined	High

Table 1. Category of decision problems.

The ITAKA research group at the URV is focused on the study and development of new methods and applications of multi-criteria decision aid systems (MCDA). They are currently working on the extension of some classical MCDA methods based on outranking relations [Gre05][Per92]. In particular, the method ELECTRE is being adapted to the treatment of non-numerical variables.

MCDA methods have as input a data matrix with a set of objects in the rows and a set of attributes (or variables) in the columns. Additionally a model of the user preferences is known so that we can evaluate the suitability of the values in the data matrix with respect to the goals (interests, preferences) of the user.

In this Master Thesis we want to study the case of semantic multi-valued attributes, as in Table 2. In this case we show 3 different activities for tourists. The activities may be described with different attributes, being one of them the “Type of activity”. Each activity has a list of concepts (tags) that define the kind of activity.

Touristic Activities	Type of activity
Gaudí museum	Art_museum, Architecture
Gaià river	Natural_space, Rural_route, Biking
Montblanc Cellar	Vineyards, Rural_route, Wine_tasting
...	

Table 2. Semantic multi-valued attributes example

When the list of available activities is large (we have catalogued more than a thousand of different activities in the Tarragona province) we need tools to automatically analyse this data from a certain user's perspective in order to recommend him only the subset of activities of his interest.

In this work we will use semantic techniques that are based on ontologies. An ontology is a knowledge resource that defines concepts and their relations. A specific touristic ontology (DAMASK ontology) is available for this work. The concepts (classes) of the ontology will correspond to the tags that can be used in the data matrix. Therefore, by exploiting the relations in the ontology we can know the similarity between tags. Ontologies can also be used to store the user's preferences, by associating some numerical scores to the concepts [Val13].

Up to now, MCDA methods are usually applied to numerical or categorical data. To deal with this new kind of semantic data we need to modify some steps of the classical algorithms. In this work we will study the ELECTRE method, which is based on the definition of so called outranking relations [Vas12][Gre05].

In parallel to MCDA methods, we should study also Semantic-based Recommender Systems. Given a set of items, the goal of a recommender system is to identify the most suitable ones according to the information stored in a user's profile by adopting diverse filtering strategies [Bla11]. Recommender systems assist in advertising tasks by automatically selecting the most appropriate items for each user as per his/her personal interests and preferences. Research in recommender systems started back in the early 1990s, but the greater advances have been due to the irruption of recent technologies like those of the Semantic Web. Semantics-based recommender systems can outperform previous recommendation approaches by exploiting two main elements:

- A knowledge base, typically an ontology, that represents semantic features or attributes of the available items.
- Filtering strategies based on semantic reasoning techniques that discover relevant relationships between the users' preferences and the items to be recommended.

The first strategies of recommender systems merely looked at demographic information (e.g. age, gender or marital status) to recommend items that had interested other users with similar data. The results so obtained tend to be imprecise and fail to reflect changes of the user preferences over time.

This problem was addressed by content-based filtering, that looks for items similar to others that gained the user's interest in the past. This strategy is easy to adopt, but bears a problem of overspecialization: the recommendations tend to be repetitive for considering that a user will always appreciate the same kind of items.

To tackle these problems, the scientific community came up with collaborative filtering, that proceeds by evaluating not only the profile of the target user (the one who will receive the recommendations), but also those of users with similar interests (his/her neighbours). This approach can solve the lack of diversity in the recommendations, but faces problems like the sparsity when the number of items is huge (which makes it hard to find users with similar evaluations for the same items) or the treatment given to users whose preferences are dissimilar to the majority (the grey sheep).

As we have seen, MCDA methods and Recommender Systems share a similar goal: support the user or users in a decision problem. While Recommender Systems deal with large and heterogeneous datasets, MCDA methods traditionally only handle small sets of numerical data. Other aspects that distinguish Recommender Systems from classical MCDA approach are that Recommender Systems are not designed for a unique decision maker (i.e., user) but they are supposed to be used at different times by many people with different interests and characteristics. Therefore, Recommender Systems must be able to manage several user profiles. Secondly, the elements of the decision making problem in a Recommender System are not static: the list of alternatives can be different each time the user asks for a recommendation and the user profile may also change over time.

In Table 3, we summarize differences between both approaches:

MCDA methods	Semantic Recommender Systems
Deal with numerical or categorical data.	Deal with semantic data.
Are designed for a unique Decision Maker.	Are designed to be used by many users with different interests.
The elements of the decision problem are static.	The elements of the decision problem are not static: alternatives and User profile can be different each time.
Use MCDA outranking methods, like ELECTRE.	Use content-based or collaborative filtering methods.

Table 3. MCDA and Recommender Sytems comparison

In summary, the objective of this master thesis would be the integration of both MCDA (originated in the field of Statistics and Operational Research) and Semantic Recommender Systems (originated in the field of Artificial Intelligence). We can consider this integration from two points of view: "Design a Semantic Recommender System that uses ELECTRE" or "Adapt ELECTRE to handle semantic data in a similar way Recommender Systems do".

The study of this kind of techniques is crucial in the development of a new generation of intelligent recommender systems, based on the interpretation of semantic data, as well as, in decision support in general.

1.1 Master Thesis Goals

The goal of this master thesis is to use the information about the user preferences and the relation of the concepts in an ontology in order to compare and order the different objects in a data matrix. The method used for comparison and ranking will be ELECTRE. This objective can be decomposed in two main sub-goals:

1. Find a suitable model to represent the user's incomplete preferences on an ontology.
2. Extend the classic ELECTRE method in order to exploit the preferences stored in the ontology for construction of a preference order.

1.2 Structure of the document

The rest of this document is organized as follows:

- Chapter 2 explains three main concepts related to the work on this master thesis. First, the definition of an ontology, because it will be the principal resource to enable the system to deal with the semantics of the information during the decision process. Second, the user profile, that stores the user preferences. Third, and last, an introduction to the techniques of the field of MCDA, focusing on the method known as ELECTRE-III.

- Chapter 3 introduces three software tools that have been used in this master thesis. Protégé is a very successful ontologies editor. Second, Diviz, regarding to availability of implementations of MCDA methods, it is currently the best system to work. Finally, Apache Jena and, concretely, its Ontology API are introduced. Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications.

- In Chapter 4, an overview of recommender systems based in the semantic representation of the user profile and the alternatives through the use of domain ontologies is presented.

- Chapter 5 presents the solution we propose in this work, named OntElectre. OntElectre is a new methodology that enables applying the standard Outranking MCDA method ELECTRE to a set of alternatives described by semantic multi-valued attributes and with a user profile based in a domain ontology.

- Chapter 6 explains a case study. OntElectre is applied to the Tourism domain, concretely to the recommendation of touristic city destinations.

- In Chapter 7, the results of the system are evaluated. We have created 4 different profiles to test the system: the Fashion Victim, the Beach Lover, the Culture Freak and the Big Family. In the four sections of this chapter the results obtained with each profile are explained in detail.

- Chapter 8 presents the conclusions of this work among the goals accomplished and the future work.

- In Chapter 9, DAMASK Ontology is explained in detail.

- Finally, chapter 10 contains the references used in this work.

2 Definition of concepts

In this work we will address the problem of the management of semantic data in decision support and recommender systems by using ontologies. In this chapter, the main concepts related to the work on this master thesis are presented.

2.1 Ontology

In philosophical contexts, ***ontology*** has traditionally been defined as the theory of what exists (or of "being qua being"): the study of the kinds of entities in reality and of the relationships that these entities bear to one another [Arp15].

In recent times use of the term ***ontology*** has become prominent in computer and information science, and has been especially successful in areas of bioinformatics. Ontologies are designed to promote greater consistency in description of data. To this end the terms in an ontology are provided with both textual definitions (to ensure consistency on the side of human beings maintaining and using the ontology) and logical definitions (to aid computer access and quality control). The result is organized in a graph-theoretic format, in which terms serve as nodes in the graph and ontological relations (as between a type and its subtypes) serve as edges.

In terms of computer science, a definition of ***ontology*** is the following [Arp15]: a representational artifact, comprising a taxonomy as proper part, whose representations are intended to designate some combination of universals, defined classes, and certain relations between them.

This definition employs a number of terms that are themselves in need of defining:

The first term ***taxonomy*** we can define as follows: a hierarchy consisting of terms denoting types (or universals or classes) linked by subtype relations.

By ***hierarchy*** we mean a graph-theoretic structure consisting of nodes and edges with a single top-most node (the root) connected to all other nodes through unique branches.

By ***types*** or ***universals*** we mean the entities in the world referred to by the nodes in a hierarchy.

2.1.1 Ontology components

Ontologies have several components on which intelligent systems may apply reasoning procedures [Val13]. The main ones are classes, instances, properties and rules. A brief explanation of these features and how they are used in some ontology-based systems is given in the following:

- ***Classes*** are the abstract representation of the different concepts of a domain. They usually correspond with the nouns found in the domain. For instance, a class could be 'city', 'accommodation' or 'singer'. Each class has a certain number of features, represented with slots. For instance, the 'singer' class could have slots identifying aspects like the birth place of the singer, his/her birth date, his/her number of Grammy awards, etc. Figure 1 shows an example of definition of a base class for the definition of a *Religious building.*

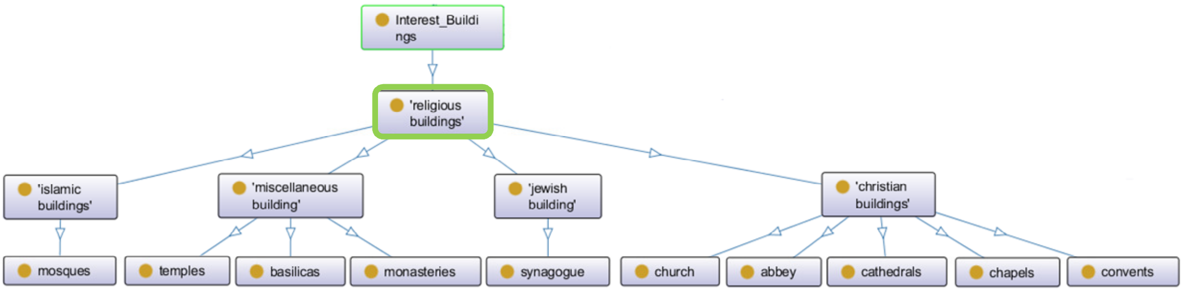


Figure 1 Class definition example

- ***Instances*** of a class represent specific individuals that belong to that class of objects. For example, in the Music domain we may have instances of the class 'singer' like 'Elton John' or 'Madonna'. In the Tourism domain, 'Berlin' and 'The Plaza Hotel' are instances of the classes 'city' and 'accommodation', respectively. Instances have a particular value associated with each of their slots, including those slots inherited from all of their superclasses.

- ***Properties*** permit to establish binary semantic relationships between classes. The most common is the 'is-a' property which indicates that a class is subclass of another class. For example, 'football' is-a 'sport' means that the 'football' class is a subclass of 'sport' (and, therefore, it inherits all its characteristics). This property defines a taxonomical structure of classes, which is normally a tree or an acyclic graph. Any other property between classes is considered nontaxonomical. For example, we could define the property 'locatedIn' between 'accommodation' and 'city', and use it to indicate that 'The Plaza Hotel' is located in 'Berlin'.

- Ontology ***rules*** are the translation of mathematical axioms that impose some constraints on the objects that can be related via a certain property or on the values that a certain slot may take. These rules may be used by ontology-based systems to implement complex reasoning mechanisms. For instance, an axiom could specify that a certain binary relationship P between classes has the transitive property; then, if the system knows that *aPb* and *bPc*, it can infer that *aPc*

2.1.2 Ontology types

Because ontologies aim at consensual domain knowledge their development is often a cooperative process involving different people, possibly at different locations. People who agree to accept an ontology are said to commit themselves to that ontology [Fen01].

Depending on their generality level, different types of ontologies may be identified that fulfil different roles in the process of building a Knowledge Base System. Among others, we can distinguish the following ontology types:

- ***Domain ontologies*** capture the knowledge valid for a particular type of domain (e.g. electronic, medical, mechanic, etc.).
- ***Metadata ontologies*** provide a vocabulary for describing the content of on-line information sources.
- ***Generic or common sense ontologies*** aim at capturing general knowledge about the world, providing basic notions and concepts for things like time, space, state, etc.
- ***Representational ontologies*** do not commit themselves to any particular domain. Such ontologies provide representational entities without stating what should be represented.
- Other types of ontology are so-called ***method*** and ***task ontologies*. **Task ontologies provide terms specific for particular tasks and method ontologies provide terms specific to particular PSMs.

2.1.3 Ontology examples

In this chapter I offer a brief explanation of some existing ontologies that are nowadays in use by the scientific community in different fields:

- ***YAGO (Yet Another Great Ontology)*:** YAGO is a knowledge base developed at the Max Planck Institute for Computer Science in Saarbrünken. It is one of the largest knowledge bases actually available composed of entities and relations. Currently, YAGO has knowledge of more than 10 million entities (like persons, organisations, cities, etc.) and contains more than 120 million facts about these entities. YAGO facts have been defined by a process which has unified two of the most important knowledge resources today: Wikipedia and WordNet. Moreover YAGO obtains information from GeoNames. The knowledge extraction process has been undertaken by using a set of advanced rule-based and heuristic methods [Suc08].

- ***FOAF (Friend Of A Friend)*:** The FOAF project was started in 2000 by Libby Miller and Dan Brickley and can be considered the first Social Semantic Web application. FOAF is a descriptive vocabulary expressed using the Resource Description Framework (RDF) and the Web Ontology Language (OWL) defining a dictionary of people-related terms that can be used in structured data. FOAF describes persons, their activities and their relations to other people and objects and allows groups of people to describe social networks without the need for a centralised database.

- ***CATO (Commercial Applications Top Ontologies)*:** CATO is a formal ontology integration engine developed to provide semantic interoperability among resources in a heterogeneous environment. It performs an automatic taxonomical alignment of ontologies. Their strategy is based on the application of well known software engineering strategies, such as lexical analysis, tree comparison and the use of similarity measurements [Bre07] [Koo05].

- ***OpenClinical (knowledge management for medical care)*:** The use of ontologies in medicine is mainly focussed on the representation and re-organization of medical terminologies. Physicians developed their own specialized languages and lexicons to help them store and communicate general medical knowledge and patient-related information efficiently. Such terminologies, optimized for human processing, are characterized by a significant amount of implicit knowledge. Medical information systems, on the other hand, need to be able to communicate complex and detailed medical concepts (possibly expressed in different languages) unambiguously. This is obviously a difficult task and requires a profound analysis of the structure and the concepts of medical terminologies. But it can be achieved by constructing medical domain ontologies for representing medical terminology systems.

- **The Gene Ontology (GO) project:** is a collaborative effort to address the need for consistent descriptions of gene products across databases. Founded in 1998, the project began as a collaboration between three model organism databases, FlyBase (Drosophila), the Saccharomyces Genome Database (SGD) and the Mouse Genome Database (MGD). The GO Consortium (GOC) has since grown to incorporate many databases, including several of the world's major repositories for plant, animal, and microbial genomes.
  The GO project has developed three structured ontologies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner.

2.1.4 OWL (Web Ontology Language)

2.1.4.1 Introduction

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. The OWL languages are characterized by formal semantics. They are built upon a W3C xml standard for objects called the Resource Description Framework (RDF). OWL and RDF have attracted significant academic, medical and commercial interest.

In October 2007 a new W3C working group was started to extend OWL with several new features as proposed in the OWL 1.1 member submission. W3C announced the new version of OWL on 27 October 2009. This new version, called OWL 2, soon found its way into semantic editors such as Protégé and semantic reasoners such as Pellet, RacerPro, FaCT++ and HermiT.

OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

- ***OWL Lite*** supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1

- ***OWL DL*** supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with *description logics*, a field of research that has studied the logics that form the formal foundation of OWL.

- ***OWL Full*** is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

2.1.4.2 OWL Syntax

Following, I enumerate most important elements in OWL Syntax:

- ***Class:*** A class defines a group of individuals that belong together because they share some properties. Classes can be organized in a specialization hierarchy using *subClassOf*. There is a built-in most general class named *Thing* that is the class of all individuals and is a superclass of all OWL classes. There is also a built-in most specific class named *Nothing* that is the class that has no instances and a subclass of all OWL classes.

- ***subClassOf:*** Class hierarchies may be created by making one or more statements that a class is a subclass of another class. For example, in Damask Ontology, the class *Canal* is a subclass of the class *Water_Landmark*. From this a reasoner can deduce that if an individual is a *Canal*, then it is also a Water_Landmark.

- ***disjointWith:*** Classes may be stated to be disjoint from each other. For example, *Man* and *Woman* can be stated to be disjoint classes. From this *disjointWith* statement, a reasoner can deduce an inconsistency when an individual is stated to be an instance of both and similarly a reasoner can deduce that if A is an instance of *Man*, then A is not an instance of *Woman*.

2.2 User profile

The representation and management of the user preferences is a key component in decision support systems (and particularly in recommender systems) because the solution must be based on the decision maker (i.e., the user) interests and needs. One of the uses of the ontology is in the definition of the user profile. Recommender systems, as other decision support systems, need to know the preferences of the decision maker.

2.2.1 Definition of the user profile

The ***user profile*** is defined as a set of information describing the user [Oua15]. It contains data that represent the user's preferences as well as their context. Different models for storing the user's profile exist in the literature of recommender systems.

2.2.2 Building the user profile

The modelling of the user first requires collecting a set of information. When considering how information can be acquired from the user, a distinction can be made between an explicit model and an implicit model. The crop of information can be:

- ***Explicit:*** by the explicit approach, users proactively communicate useful information to the system, e.g., by compiling questionnaires [Ali14]. However, it has been shown that in practice few users are taken to make the effort to provide quality information and keep them updated [Oua15].

- ***Implicit:*** user interests are automatically inferred based on the analysis of the actions they perform, for example, during Web search (e.g., how users move from one Web page to another, or which documents they download, etc.) [Ali14]. To the aim of defining a user profile the following activities are usually undertaken:

    1. Individuating and collecting the knowledge related to the user's preferences and interests.
    2. Selecting a formal language to represent the collected knowledge. Ontologies have been more recently considered as a powerful expressive means for representing user profiles. The advantage offered by ontological languages is that they allow a more structured and expressive knowledge representation with respect other approaches (bags of words, vectors or conceptual taxonomies)
    3. Defining a strategy for updating the profile.

- ***Hybrid:*** Sometimes it is claimed that requiring so much information by means of forms is not appropriate because many users will abandon the system even before starting to use it [Val13]. Hybrid approaches are used to alleviate this effect. For example one may use information about the user context to infer some of the data, given that the personal characteristics determine the human behaviour and the behaviour determines the context, and vice versa. In [Jun13], for instance, they use explicit information (demographic data) and implicit information (TV program contents usage history) to determine the user profile.

2.2.3 Initialization of the user profile

We usually distinguish two phases in the modelling of the profile: initialization and update [Oua15]. First, the profile is initialized on the first use of the system. There are four main strategies:

- ***The empty boot***: the profile does not contain information. It is then grown over the use of system

- ***Manual initialization***: the user is asked to complete a form in which the information will be exactly the content of the profile.

- ***Semi-automatic initialization***: the user is asked to complete a form and/or answer a set of questions. These responses allow the system to build an initial profile. Unlike the previous strategy, the information provided by the user are not forming directly the profile, it is from these that the profile is derived.

- ***Automatic initialization (implicit)***: The system automatically deducts the profile from a set of previously stored data. For example, some systems initialize profile from Web browser history, others use the files in the user's home directory. Adaptation can be based on explicit or implicit feedback. Explicit feedback requires the user to evaluate examined items on a scale. In implicit feedback the user‘s interests are inferred by observing the user‘s actions, which is more convenient for the user but more difficult to implement.

2.2.4 Update of the user profile

The information need of a user on different topics normally changes over time [Oua15]. In addition, the interests of a user are not always known beforehand such as in the case of implicit modelling. The user profile must therefore be able to adapt to changes in the user‘s actual interests. There are three ways to update the profile:

- ***Manual Update***: the user is responsible for updating his profile when necessary. This kind of update is generally associated with manual initialization. In practice the user hardly ever change his profile because he usually is not aware of the interest that this expensive task can be compared to its immediate purpose.

- ***Update via explicit feedback***: this strategy is to give the opportunity to the user to explicitly give its judgment on the results returned by the system. This judgment is generally expressed either as a Boolean (satisfied/not satisfied; or like/not like), as a note or as a text comment. But users little or not respond at all to requests of explicit feedback because they do not see the direct benefit they could get by making the effort to answer such questions.

- ***Update via implicit feedback***: is to retrieve implicitly (without prompting the user) the judgment of the user on the results returned by the system. To do this, feedback is deduced based on the observation of user behaviour. In [Oua15], the authors presented web action examples of implicit feedback and the domain in which they are used (see Table 4)

Action	Description	Domain example
Purchase (Price)	buys item	Commercial web site
Assess	evaluates or recommends	Recommender system
Repeated Use (Number)	action is repeated	Web site
Save/Print	saves item to personal storage	email
Delete	deletes item	email
Refer (Time)	cites or otherwise refers to item	Usenet News
Reply (Time)	replies to item	Usenet News

Mark	adds to an "interesting" list	Web browsing
Examine / Read (Time)	looks at whole item	Web site
Consider (Time)	looks at abstract	Web site
Glimpse	sees title in a list	Web browsing
Query	asks for information	Web site

Table 4. Potential types of implicit feedback

Commercial web site for example uses purchase information from users to make recommendations. When the entry of a book is displayed other book titles are shown which were purchased by customers who also bought the selected book. The time that a user spends on reading an article could also be considered as implicit feedback. The problem with this approach is that it is not possible to determine whether the user is actually reading the article or has taken a break.

2.3 MCDA AND ELECTRE

We present in this section an overview of MCDA methods and their different approaches. After, we will explain in detail ELECTRE method, concretely ELECTRE III as this is the outranking method used by the proposed methodology in present work.

2.3.1 Multi Criteria Decision Analysis (MCDA)

The growing recognition that decision makers will often try to achieve multiple, and usually conflicting, objectives has led during the last three decades to the development of multi-criteria decision analysis (MCDA). This field started in Business and Economy faculties together with researchers from Statistics and Operational Research disciplines. This is now a vast field of research, with its scientific research community and its specialized journals[1]. A large and growing number of real-world applications are continuously developed, mainly for supporting both public policy making and decisions by private corporations.

Decision problems such as ranking, choice and sorting problems are often complex as they usually involve several criteria. People no longer consider only one criterion when making a decision. Most of the time, there is no one, perfect option available to suit all the criteria: an "ideal" option does not usually exist, and therefore a compromise must be found.

To address this problem the decision maker can make use of naïve approaches such as a simple weighted sum. Multi-criteria decision analysis (MCDA) methods have been developed to support the decision maker in their unique and personal decision process. They have the distinction of placing the decision maker at the centre of the process [Gre05]. They are not automatable methods that lead to the same solution for every decision maker, but they incorporate subjective information. Subjective information, also known as preference information, is provided by the decision maker, which leads to the compromise solution.

[1] EURO group EWG-MCDA: www.cs.put.poznan.pl
Int. Society MCDM: www.mcdmsociety.org

2.3.1.1 Decision problems classification

Three main types of decision problems have been identified according to the goal to achieve [Gre05]:

1. **The choice or selection problem**. The goal is to select the single best option or reduce the group of options to a subset of equivalent or incomparable 'good' options.
2. **The classification or sorting problem**. Options are sorted into ordered and predefined groups, called categories. The aim is to then regroup the options with similar behaviours or characteristics for descriptive, organizational or predictive reasons.
3. **The ranking problem**. Options are ordered from best to worst by means of scores or pairwise comparisons, etc. The order can be partial if incomparable options are considered, or complete.

2.3.1.2 MCDA Approaches

Multi Criteria Decision methods can be classified into three groups, according to their approach [Ish13]:

- **Full aggregation approach (or American school).** A score is evaluated for each criterion and these are then synthesized into a global score. This approach assumes compensable scores, i.e. a bad score for one criterion is compensated for by a good score on another. Belong to this approach methods like MAUT, AHP, OWA.
- **Outranking approach (or French school).** A bad score may not be compensated for by a better score. The order of the options may be partial because the notion of incomparability is allowed. Two options may have the same score, but their behaviour may be different and therefore incomparable. The most well-known outranking models are: ELECTRE and PROMETHEE
- **Goal, aspiration or reference level approach**. This approach defines a goal on each criterion, and then identifies the closest options to the ideal goal or reference level.

2.3.2 ELECTRE

The ELECTRE (ELimination Et Choix Traduisant la REalité) method was designed in France in the late 60s by Bernard Roy and belongs to the outranking methods [Roy68]. The outranking methods are based on pair wise comparison of the options. This means that every option is compared to all other options.

The main characteristic and advantage of the ELECTRE methods is that they avoid compensation between criteria and any normalization process, which distorts the original data. They can be subdivided according to the type of problem they solve: Choice problem, Ranking problem, Sorting problem.

ELECTRE methods aim at building a binary outranking relation S, where *a* S *b* means *"a is at least as good as b*".

ELECTRE methods establish a realistic representation of four basic situations of preference: strict preference, weak preference, indifference, and incomparability.

Considering two alternatives, *a* and *b*, the four basic situations are defined as follows:

- ***Strict preference (a P b)***: it corresponds to a situation where there are clear and positive reasons in favour of one of the two alternatives.
- ***Weak preference (a Q b)***: it corresponds to a situation where there are clear and positive reasons that invalidate strict preference in favour of one of the two alternatives, but they are insufficient to deduce either the strict preference in favour of the other alternative or indifference between both alternatives, thereby not allowing either of the two preceding situations to be distinguished as appropriate.
- ***Indifference (a I b)***: it corresponds to a situation where there are clear and positive reasons that justify an equivalence between the two actions.
- ***Incomparability (a R b)***: it corresponds to an absence of clear and positive reasons that would justify any of the three preceding relations.

ELECTRE methods have been widely acknowledged as effective and efficient decision aiding tools, with successful applications in different domains.

2.3.3 ELECTRE III

ELECTRE III is the most used ranking method in the ELECTRE family. Ranking methods may lead to a partial order on a set of options but without assigning a score to the alternatives. The preference order amongst the options is the output of the method.

ELECTRE III is divided into two phases. First, the outranking relationship between the options is constructed and then exploited. Most of the information from the decision maker is required in the first phase: the weight of the criteria, the indifference, the preference and the veto thresholds.

ELECTRE III makes use of outranking relations. An outranking relation, where *a outranks b* (denoted by *a S b*), expresses the fact that there are sufficient arguments to decide whether *a* is at least as good as *b* and there are no essential reasons to refute this [Roy74]. An outranking degree *S(a,b)* between *a* and *b* will be computed in order to 'measure' or to 'evaluate' this assertion.

The strength of the assertion *a outranks b* is given by the credibility or outranking degree *S(a,b)*. It is a score between 0 and 1, where the closer *S(a,b)* is to 1, the stronger the assertion. This outranking degree considers two perspectives: the concordance and the discordance of the statement that *a* outranks *b.* The concordance and discordance are measured respectively while incorporating the decision maker's preference on various criteria. The DM is required to provide the indifference and preference thresholds for calculating the concordance degree, while the veto threshold is needed for the discordance degree.

Briefly, ELECTRE III main steps are as follows:

1. Building the outranking relation:

- Measure the concordance condition: a majority of criteria must support *a S b* (classical majority principle).
- Measure the discordance condition: among the non concordance criteria, none of then strongly refutes *a S b* (respect of minorities principle).
- Calculate a credibility index, which will correspond to the Outranking relation for each pair of alternatives, *a S b.*

2. Exploiting the outranking relation:

- Outranking relations are represented by a graph.
- The outranking relation is used to solve the decision problem.

2.3.3.1 Concordance

A partial concordance degree $c_i(a,b)$ measures the assertion '*a outranks b*' or '*a is at least as good as b*' on the specific criterion g_j.

The uncertainty or indifference of the DM preference model can be represented with the following intra-criteria parameters:

- The indifference threshold q_j, below which the DM is indifferent to two alternatives in terms of their performances on criterion g_j.

- The preference threshold p_j, above which the DM shows a clear strict preference of one alternative over the other in terms of their performances on criterion g_j.

Between these two thresholds, the partial concordance degree is computed on the basis of a linear interpolation, represented in Figure 2.

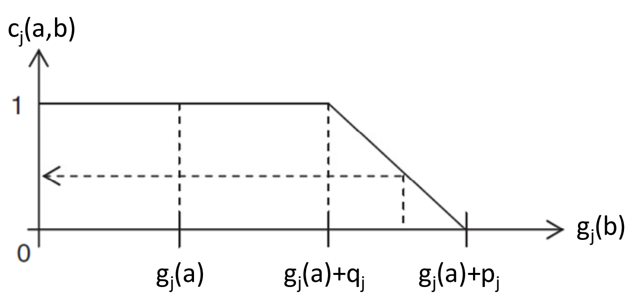


Figure 2. The partial concordance index $c_i(a,b)$.

Thus, the partial concordance degree can defined as follows:

$$c_j(a,b) = \begin{cases} 1 & if\ g_j(a) \geq\ g_j(b) - q_j \\ 0 & if\ g_j(a) \leq\ g_j(b) - p_j \\ \dfrac{p_j -\ g_j(a) -\ g_j(b)}{p_j - q_j} & otherwise \end{cases}$$

A global concordance degree *C(a,b)* aggregates all the partial concordance indexes on the different criteria by taking into account their corresponding criteria weight. This global index is thus the weighted sum of all the partial concordance indexes and measures how concordant the assertion *'a is at least as good as b'* is regarding all the criteria:

$$C(a,b) = \frac{1}{W}\sum_{j=1}^{m} w_j\, c_j(a,b)$$

Where w_j is the importance of attribute j and *W* is the total importance of all attributes.

2.3.3.2 Discordance

On the other hand, the partial discordance degree $d_i(a,b)$ measures the decision maker's discordance with the assertion *'a is at least as good as b'* on criterion g_j. If the decision maker, when considering criterion g_j, strongly disagrees with the assertion, the discordance degree reaches its maximum value 1 and reflects the fact that g_j sets its veto. This is the case if the difference in performances (i.e. $g_j(b)$ - $g_j(a)$) is higher than a so-called veto threshold, denoted by v_j. The discordance degree has minimum value 0, when there is no reason to refute the assertion. As in the case of the partial concordance degree, between these two extremes, $d_i(a,b)$ will vary linearly between the preference and veto thresholds as a function of the difference $g_j(b)$ - $g_j(a)$. The partial discordance index is defined as follows:

$$d_j(a,b) = \begin{cases} 0 & if\ g_j(b) \leq\ g_j(a) +\ p_j \\ 1 & if\ g_j(b) \geq\ g_j(a) +\ v_j \\ \dfrac{g_j(b) -\ g_j(a) -\ p_j}{v_j - p_j} & otherwise \end{cases}$$

If no veto threshold v_j is specified $d_i(a,b)=0$ for all pairs of alternatives.

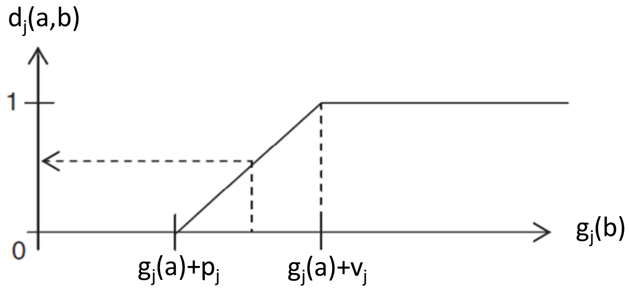


Figure 3. The partial discordance degree $d_j(a,b)$

2.3.3.3 Outranking degree

Finally, a global outranking degree *S(a,b)* summarizes the concordance and discordance degrees into one measure of the assertion *'a outranks b'* using the following formula:

$$S(a,b) = \begin{cases} C(a,b) & \text{if } d_j(a,b) \le C(a,b) \forall j \\ C(a,b) \displaystyle\prod_{d_j(a,b) > C(a,b)} \frac{1 - d_j(a,b)}{1 - C(a,b)} & \text{otherwise} \end{cases}$$

If no veto threshold v_j is specified *S(a,b) = C(a,b)* for all pairs of alternatives.

2.3.3.4 Distillation

The second phase consists of exploiting these pairwise outranking degrees: the *ascending* and *descending* distillation procedures lead to a complete pre-order. Each pre-order takes into account respectively the outranking and outranked behaviour of an alternative with regard to the others. Since these procedures may lead to two different procedures, a final partial pre-order is generated as the intersection of the ascending and descending pre-orders. In this partial pre-order we can distinguish 3 types of relations: preference, indifference and incomparability. Afterwards a ranking may be obtained from it by following these different relations.

2.4 Summary

In this chapter three main topics have been presented in detail. First, the definition of an ontology, because it will be the principal resource to enable the system to deal with the semantics of the information during the decision process. Second, to give a personalized answer to the decision maker, the importance of the user profile has been presented, addressing different topics related to the initialization and update of the preference information stored in the user profile. Third, and last, an introduction to the techniques of the field of MCDA (Multiple Criteria Decision Aiding) has been done, focusing on the method known as ELECTRE-III, as this is the one that will be used in this work.

3 Tools

This chapter explains three software tools that have been used in this master thesis. Protégé is a very successful ontologies editor. It has been chosen for its popularity, facility of use and support of OWL and RDF language among others.

Regarding to availability of implementations of MCDA methods, Diviz is currently the best system to work. It is a free platform based on web-services that provides access to the most well-known MCDA methods in the literature in an easy way. It has been used previously in ITAKA research group, who has also contributed to develop some of the available components in Diviz.

Finally, we will also mention Apache Jena and, concretely, its Ontology API. Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications. The framework is composed of different APIs interacting together to process RDF data. This API has been chosen because it is widely-extended and well documented. In has already been used in ITAKA research group with success.

3.1 Diviz and Decision Deck project

The Decision Deck project is a French non profit association that aims at collaboratively developing Open Source software tools implementing Multi-Criteria Decision Aiding techniques which are meant to support complex decision aiding processes. One of the main features of these software solutions are that they are interoperable in order to create a coherent ecosystem.

The Decision Deck Consortium was officially born on November 25, 2008.

Some of Decision Deck project initiatives are:

- **Diviz**: A software for designing, executing and sharing MCDA methods, algorithms and experiments
- **XMCDA**: A standardized XML recommendation to represent objects and data from the MCDA field which allows MCDA software to be interoperable
- **XMCDA web services**: Distributed computational MCDA resources, using the XMCDA standard
- **GIS**: An initiative linking Geographical Information Systems and MCDA

3.1.1 Diviz

The diviz software is a tool for designing, executing and sharing Multiple Criteria Decision Aid (MCDA) techniques. The target user community of diviz reaches from researchers who require to construct algorithmic workflows of elementary MCDA components to teachers who need to present MCDA methods and let their students experiment them. Besides, one of the principal features of diviz is to ease the comparison of results issued from different methods on the same decision problem. This allows among others to check the robustness of the output of a decision aid process with respect to the choice of the decision aid technique.

The goals of diviz are:

- to help researchers to construct algorithmic MCDA workflows from elementary MCDA components;
- to help teachers to present MCDA methods and let the students experiment their own creations;
- to help to easily compare results of different methods;
- to allow to easily add new elementary MCDA components;
- to avoid heavy calculations on your local computer by executing the methods on distant servers;
- to ease the dissemination of new MCDA algorithms, methods and experiments.

Based on basic algorithmic components, diviz allows combining these elements in view of creating complex MCDA workflows or MCDA methods. Once the workflow is designed, it can be executed on various data sets written according to the XMCDA standard. This execution is performed on distant servers via web services. In Figure 4 we have a diviz screen shot with their components.

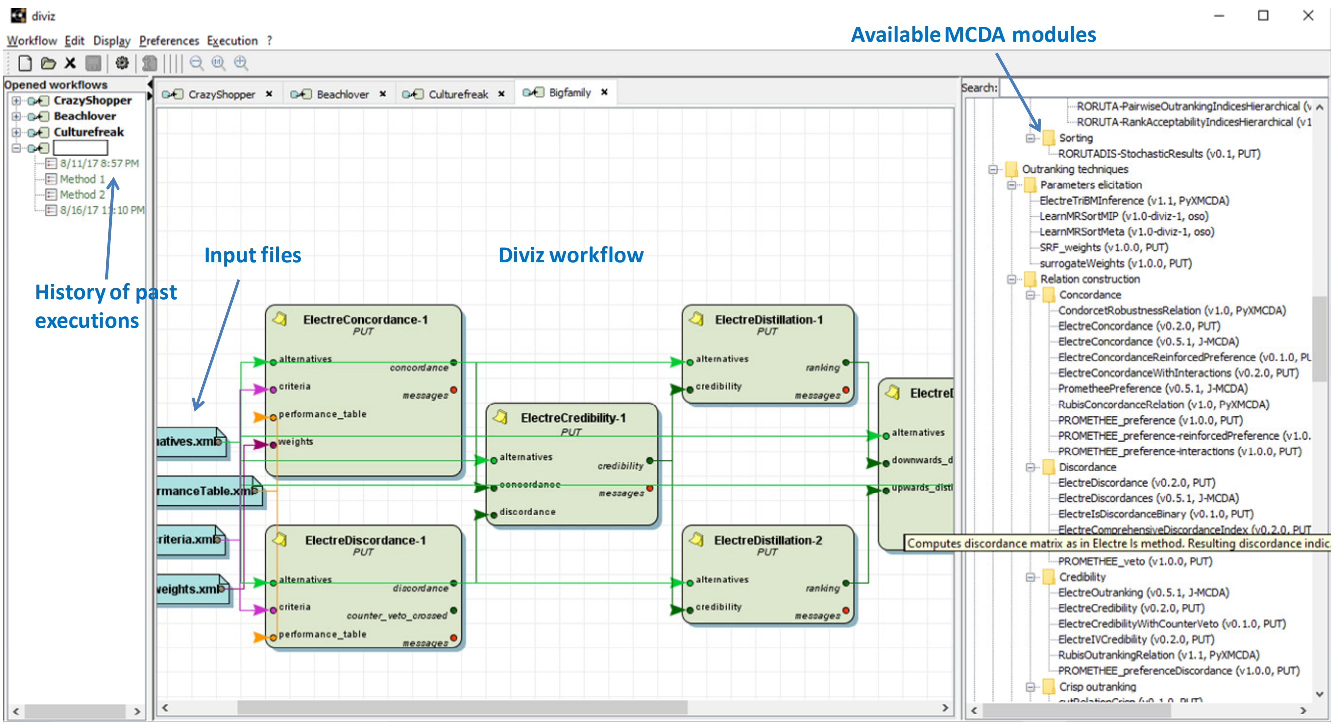


Figure 4. Diviz components

Once the execution is completed, the outputs of the different elementary components are available and can be visualised in diviz. The user can then modify the input files in order to rerun the workflow, or change the existing workflow to adapt it to his needs. The history of past executions is available at any moment, which helps to compare the input and output data and tune the parameters or the workflow.

Finally, the constructed workflow can be deployed as a new web service which is then automatically made available to all the users which are employing diviz (in development).

The data exchanged between the elementary algorithmic components is conform to the XMCDA standard. XMCDA allows expressing MCDA concepts through a few general XML structures.

3.1.2 XMCDA

All diviz files are written in XMCDA. XMCDA is a data standard which allows representing MCDA data elements in XML according to a clearly defined grammar.

XMCDA is an instance of UMCDA-ML, which is the Universal MultiCriteria Decision Analysis Modelling Language and which is one of the scientific initiatives inside the Decision Deck project. UMCDA-ML is intended to be a universal modelling language to express MCDA concepts and generic decision aid processes.

XMCDA focuses more particularly on MCDA concepts and data structures and is defined by an XML schema. The goals of XMCDA are to ease:

- the interaction of different MCDA algorithms.
- the execution of various algorithms on the same problem instance.
- the visual representation of MCDA concepts and data structures via standard tools like web browsers.

3.2 Protégé

Protégé was developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. It is a free, open-source ontology editor and framework that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies.

It is actively supported by a strong community of users and developers and fully supports the latest OWL 2 Web Ontology Language and RDF specifications from the W3C. Protégé is based on Java, it is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

Protégé can be used in either a hosted solution at http://webprotege.stanford.edu or it can be installed on user's own server. It provides the following features:

- Support for editing OWL 2 ontologies.
- A default simple editing interface, which provides access to commonly used OWL constructs.
- Full change tracking and revision history.
- Collaboration tools such as, sharing and permissions, threaded notes and discussions, watches and email notifications.
- Customizable user interface.
- Customizable Web forms for application/domain specific editing.

- Support for editing OBO ontologies.
- Multiple formats for upload and download of ontologies (supported formats: RDF/XML, Turtle, OWL/XML, OBO, and others).

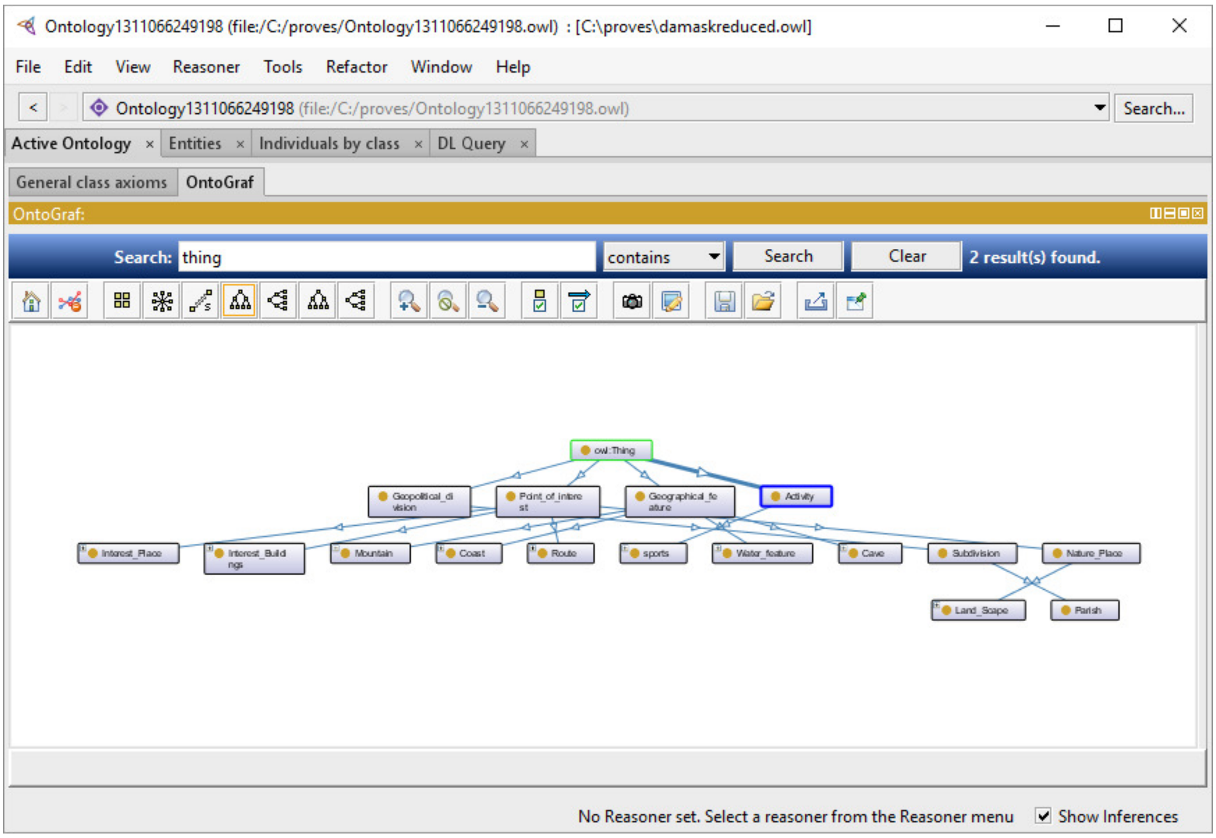


Figure 5. Protégé: Ontology editor.

Although Protégé has a wide list of features, I have only used it to edit DAMASK Ontology. All Ontology images in this work have been taken with Protégé. I have also used it to convert the original Ontology to a format readable by Jena API.

3.3 Apache Jena - Ontology API

Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications. The framework is composed of different APIs interacting together to process RDF data. We have used Ontology API in OntElectre App to manage Ontology classes and properties.

The Jena Ontology API is language-neutral: the Java class names are not specific to the underlying language. For example, the *OntClass* Java class can represent an OWL class or RDFS class. To represent the differences between the various representations, each of the ontology languages has a profile, which lists the permitted constructs and the names of the classes and properties. The profile is bound to an *ontology model*, which is an extended version of Jena's Model class

In Table 5, we summarize Jena packages used in the implementation of the proposed methodology.

Apache Jena package	Description
org.apache.jena.ontology	Provides a set of abstractions and convenience classes for accessing and manipulating ontologies represented in RDF.
org.apache.jena.rdf.model	A package for creating and manipulating RDF graphs.
org.apache.jena.util	Miscellaneous collection of utility classes.
org.apache.jena.util.iterator	A package for defining useful iterators and iterator operations, including concatenation, mapping, filtering, empty and singleton iterators, iterator wrappers, and the ExtendedIterator class used in many places in Jena.

Table 5. API Jena Packages used.

4 Background and related work

We present in first part of this section an overview of recommender systems based in the semantic representation of the user profile and the alternatives through the use of domain ontologies. Most of these works focus on the user profile initialisation and maintenance. After, they apply techniques of semantic similarity and collaborative filtering to generate the recommendation list for the user.

Finally, in chapters 4.2 and 4.3 of this section, we present two previous works done in the URV that apply outranking MCDA methods to make recommendations [Bor12][Mar16], concretely ELECTRE.

4.1 Background on User Profiles with ontologies

Several studies have been done to represent the user profile (i.e. user preferences) with ontologies, some of them have gone a step further using also ontologies in the alternatives representation and as base of recommender systems methodology. This is the case of [Nau08] that is fully based on ontologies, which are used to formalize both the user and his/her interests and the audiovisual content. Other approaches integrate ontology as an additional feature to improve classical techniques.

Ten Recommender Systems have been found in a search in the literature and they have been compared to develop the present background section. The field of application is diverse. In two systems they do TV program recommendation [Jun13][Nau08]. [Pin07] and [Bla11] are online shopping recommenders. [Bor14] and [Bat12] are touristic recommenders, in case of [Bat12] they are specialized in elderly people travelling. In [Cal13] they present a method to automatically define a personal ontology used to web search recommendation. Similarly, in [Ely14] they present a method to build personal ontologies to recommend Human-Computer interfaces. Other specific examples are [Ali14] being a "Career pathway recommendation in e-learning environment" and [Ajm13], that is a custom recommender.

In some cases the authors define their own domain ontology and in others they use existing ones like YAGO, FOAF or geo-location. It is also interesting the analysis of hybrid or implicit methods to initialize and maintain the user profile. In [Bla11] they define time functions built from the consumption stereotypes, so they base recommendations in the combination of the user profile and these time functions. In [Ajm13] they obtain the user profile explicitly with the analysis of the body colours and shapes of the user.

None of the systems studied use MCDA techniques, most of them use Collaborative filtering or hybrid techniques. We take as example [Pin07], in their work; the authors propose a hybrid approach combining semantic similarity with collaborative filtering to generate recommendation lists for users. In [Pin07] the active user u_i's personalized recommendation list is taken from its neighbourhood by deriving the top-N recommendations. The methods often adopted for recommendation generation include *Most-frequent-item* recommendation, *Association-rule-based* recommendation, *weighted-average-of-ratings* method and *deviation-from mean* method. The authors opt for the last method for it performs well in the situation where rating data density of different users differs much with each other.

Table 6 shows a comparison of the principal features of the Semantic Recommender Systems analyzed. Although the final domains are varied, we can observe that all of them are related to web consumption. This fact is comprehensible as the amount of information in the web is so huge that users need a guide to choose.

Another fact we can observe is that only 2 of the systems analyzed construct the user profile in an implicit way, the others need at least a basic profile information provided by the user.

MCDS	Domain	Ontology	User Profile
[Pin07]	Online shopping	Authors	Implicit
[Cal13]	Web Search	YAGO	Implicit
[Bla11]	Online shopping	Authors	Implicit
[Jun13]	TV Program	Authors	Hybrid
[Ely14]	UI ergonomics	HCI	Explicit
[Ali14]	Career Pathway in e-earning	Authors	Explicit
[Nau08]	Mobile TV	TVA	Hybrid
[Bat12]	Elderly People Travel	EIFFEL, FOAF	Explicit
[Ajm13]	Fashion	Authors	Explicit
[Bor14]	Tourism	Sigtur, geo-location	Explicit or Hybrid

Table 6. Recommender systems with ontology profile comparison

4.2 Ontology-based management of uncertain preferences in user profiles

In his PhD Thessis, Dr.Borràs designed and developed a recommender system to provide personalised recommendations of touristic activities in the region of Tarragona. This system applies content-based recommendation techniques using ontological information about the Tourism domain [Bor12][Bor15].

In collaboration with the experts of Parc Científic Tecnològic de Turisme i Oci (Vilaseca, Catalonia), they designed a fuzzy approach to store the user profile in a ontology, considering that each concept in the ontology is a fuzzy set and any user belongs to this fuzzy set to a certain degree. To represent the uncertainty on this information, a degree of confidence on the membership value was also included.

For each user u, $\mu_c(u)$ gives the membership degree of u to a concept c that belongs to the ontology. This membership degree is personal for each user and represents his/her degree of interest in a certain concept c. If the user is completely interested in c, then $\mu_c(u)$=1. Oppositely, when $\mu_c(u)$=0, we assume user u is not interested at all in concept c.

In order to discover the user's preferences, the application acquires both explicit and implicit information from him/her. The former includes the specification of the travel motivations and the rating values given by explicit evaluations of items. The latter is obtained from the observation of the actions of the user on the system. User preferences are only expressed in leaves.

After an initial assignment of preferences, a spreading algorithm is used to propagate the preference values of the ontology nodes to their ancestors. This process has two steps: upwards propagation (in which the interests on the ancestors of the modified leaves are updated) and downwards propagation (in which the preference and confidence on other descendants of these ancestors are also updated).

A confidence level $CL_c(u)$ is maintained for each node. This confidence level has values between 0 and 1 that quantify the confidence associated to the estimation of the membership degree of *u* to the concept *c*, denoted as $\mu_c(u)$. A large value of $CL_c(u)$ indicates that we can trust the value of $\mu_c(u)$ as the true degree of interest of the user *u* for the concept *c*, whereas a low value indicates that the estimation is not so reliable. In that way, not only the degree of membership to the concepts in the ontology is considered to select the best alternatives, but also the confidence on the estimation of those values is taken into account. For instance, the recommender system may decide to ignore the values with a low confidence level, because they have not achieved enough support. The overall interest score is obtained with the OWA aggregation taking into account the membership degree and the confidence level. This procedure is done to find the value of the Content Similarity criterion.

In order to make the final suggestion of the best alternatives, in SigTur/E-Destination recommender system, they use ELECTRE outranking method and consider up to 8 criteria to make the final proposal of activities to each user: Content similarity, Collaborative similarity, location, budget, kids, calendar, language and promotion. Content similarity criterion, corresponds to an overall interest score for a certain alternative.

4.3 Construction of an outranking relation based on semantic criteria with ELECTRE-III

In her work Martínez-García, Valls and Moreno proposed an enhancement of ELECTRE-III to deal with alternatives defined on semantic criteria [Mar16]. First, they proposed a mechanism by which the preference scores missing in a semantic user profile may be estimated from the known ones, using the taxonomical relationships of the ontology. Second, they presented a method that constructs an outranking relation on a set of alternatives defined on semantic multi-valued criteria using a new definition of the concordance and discordance fuzzy functions. This relation is later exploited with standard ELECTRE methods to rank the alternatives.

The semantic user profile model proposed in this work consists on adding a numerical property to each of the most specific concepts of the ontology (i.e. the leaves of the ontological tree), called Tag Interest Score (TIS). This value represents the degree of interest of the decision maker in the corresponding concept, with a satisfaction score in [0,1].

Initially, they know the scores of some leaf concepts, which may have been obtained explicitly or implicitly from the user. After, the system estimates unknown scores for the rest of leaves in the ontology. To estimate the TIS of concept *c*, they propose following the taxonomic relations to find scored concepts that are semantically similar to *c.* A set of leaf concepts is built by following the taxonomical relations in the ontology using *relatives(c,l)*, where *l* is the number of is-a levels we want to explore. The function *leaves(t)* returns the set of leaves of the ontology subtree whose root is *t*.

$$relatives(c,l) = \begin{cases} leaves(father(c)) & l = 1 \\ relatives(father(c), l-1) & l > 1 \end{cases}$$

They propose to start with *l*=1. If the number of elements with a known score is below a given threshold, they move to *l*=2 and so on.

The value of TIS(c) is calculated using WOWA averaging operator on the known scores given to the relatives of c. They give a different importance to each of the values that are aggregated in terms of the number of levels used to find it (so that for instance concepts obtained in level 1 are given more relevance than concepts found in level 2).

They define a new concordance and discordance based in the Semantic Win Rate $SWRj$ (a,b). It is a numerical value in [0,1] that indicates the degree of performance of alternative *a* with respect to *b* on the semantic criterion g_j. It is based on the two sets of tags $g_j(a)$={$t1,a,t2,a,t3,a,…,t|gj(a)|,a$ } and $gj(b)$={$t1,b,t2,b,t3,b,…,t|gj(b)|,b$ } and it is calculated by counting the number of times that the TIS scores of the alternative *a* are better than the ones of alternative *b.* The final value is normalized so it represents a proportion or rate.

Thus, $SWRj$ (a,b) is the percentage of pairwise comparisons between the tags of *a* and *b* for the semantic criterion gj for which the user has a higher (or equal) preference for the *a*-tag than for the *b*-tag. Using this value, the ELECTRE-III partial concordance and partial discordance indices are redefined in terms of the SWR value instead of using the standard procedure based on two performance scores. This approach permits to handle the semantic multi-values criteria following a pairwise model as the usual in outranking methods, in order to avoid compensation and allow a qualitative comparison of the tags. New thresholds are proposed then to manage the uncertainty in the concordance and discordance calculations. These thresholds are defined also in terms of the Semantic Win Rate between two alternatives.

4.4 Summary and discussion

We have presented in first part of this section an overview of Semantic Recommender System, most of these works focus on the user profile implicit or hybrid maintenance. It is not the goal of present work to design an implicit method to obtain the user profile, but it could be an interesting line of future work.

After, we have presented two works that adapt ELECTRE to handle with semantic attributes. Both explore the ontology to determine the similarity between user preferences and the alternatives. However, in both cases preferences are expressed in the ontology's leaves. This restriction, could imply a high cost of computing and maintenance of a high quantity of preference values, because the number of leaves is usually very large in a ontology.

In case of Dr.Borràs work, the semantic data grouping in a single criterion could cause compensation and loss of information.

5 Proposed methodology

As we have seen in previous works, most of the Recommender Systems using ontologies for the definition of the user profile studied use Collaborative methods. These techniques do recommendations following the principle of: "Other people that liked the same as you, also liked...". A major inconvenient of these techniques is the effect *grey sheep*, which recommends alternatives that are too general, discarding exclusive characteristics of the DM.

On the other hand, we have also seen two works that apply outranking MCDA methods to make recommendations [Bor12][Mar16]. These methods share an interesting idea of representing the uncertainty under the ontology morphology, thus the more distance between classes, the less certainty we have in the value inferred. Nevertheless, both works also share a main handicap: they summarize the semantic preferences in just a single criterion, which is later integrated with other types of criteria. Moreover, due to the fact that they try to infer unknown values, the computation cost would be too high to deal with big amounts of data.

The solution we propose in this work, named OntElectre, simplifies computing time and converts a semantic problem into a real multi-criteria numerical problem that can be solved with any of the existing outranking MCDA methods without needing any additional data analysis technique. Moreover, semantic criteria will have the same treat as other numerical criteria, like price or number of habitants, consequently both type of criteria maybe exploited together.

In Figure 6 we can see an overview of a MCDA process steps. The proposed methodology focuses in first and second steps: *User preferences modelling* and *Alternatives performance evaluation.*

For steps three and four, we can use existing and widely-acknowledged methods existing in MCDA field. In this work, we have chosen the outranking method ELECTRE III with Distillation (presented in section 2.3).

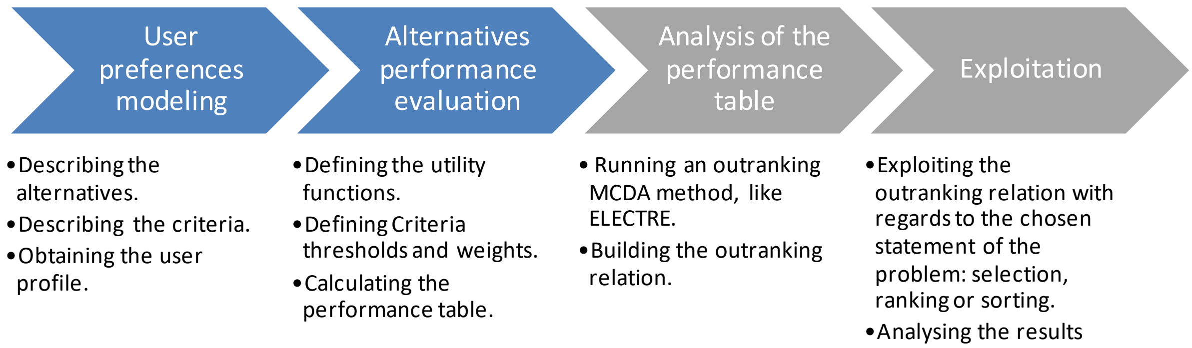


Figure 6. MCDA steps

In Table 1 we saw a classification of decision problems from the point of view of the nature of the decision. OntElectre is oriented to solve operational and tactical problems, thus it is expected a high level of automation without requiring much DM interaction.

Let us take as an example of operational decision problem, a situation in which the DM may decide a film to watch on Netflix on the weekend. Suppose we start asking him/her an interview about if he/she likes action films, or romantic films, or fantastic ones, and so on, and then we ask the level of liking for each category: do you like it a lot, only a few, in what percentage?. And after, we ask what is the difference between each category he/she would accept as indifferent. And we keep on asking questions. Finally, probably, the DM will say: "Please, give me the films list and I will decide by myself". By the other hand, the result is not crucial at all: if the DM does not like the film chosen he/she can change. It is clear that, for operational decision problems we should apply totally automatic processes, including the user profile maintenance.

In the opposite side, an example of strategic decision problem could be the purchase of a house. This is a decision the DM will probably take no more than three times in his/her live. Consequently, the expected DM involvement in this process would be much higher and the result would also be crucial for the DM.

Our case study, the decision of a tourism destination, may be situated in the middle: it could be considered a tactical problem. The decision will be probably taken few times on a year, so we can expect a medium level of DM implication and the result is not trivial but neither crucial for the DM.

OntElectre is a recommendation technique for semantic data that uses the MCDA method ELECTRE during the analysis and decision making process. Although OntElectre takes as inspiration from previous works, it has some distinguishable features with respect to them. First, OntElectre uses a representation of semantics of the concepts taking into account the ontology topology, but without needing the definition of additional concepts or calculations. Outranking methods, and concretely ELECTRE, already have mechanisms to represent the uncertainty. In ELECTRE uncertainty is modelled by the indifference and preference thresholds. So, in this work we propose to use these existing mechanisms to represent the uncertainty and, additionally, the method proposed will find the value of these thresholds automatically, instead of giving to the DM the responsibility of defining them.

Another important difference with other works is that in OntElectre the system only evaluates known information without trying to infer unknown values, this inference could fall in a dangerous compensation and would increase exponentially the chances of those alternatives with a lot of values in one aspect but that are not good in other ones.

The main key of OntElectre consists on selecting some of the classes (i.e. concepts) of the ontology and transform them into evaluation criteria. For example, let us consider that *Beach*, *Clothes shop* and *Commercial building* are the criteria that the DM uses to find a suitable city to visit. The classes that descend for one of the selected concepts will be the possible values used for the assessment of the criterion.

In this regards, we can define the decision problem that we will study as a problem of ranking a list of alternatives using semantic information. In particular each alternative will be described with a set of tags that correspond to concepts of a reference ontology. See the example in Table 7.

Alternative	Semantic information
Dijon	Motor_Sport#Formula_One#Basketball#Football#Church#Cathedral#House#Palace#Fair#Museum#Botanical_Garden#Crypt#University
Tallinn	Ice_Hockey#Church#Cathedral#House#Hotel#Palace#Tower#Open_Air_Museum#Art_Museum#Art_Gallery#Modern_Art_Museum#River#Statue#University#Library
Inverness	Football#Rugby#Cricket#Tennis#Church#Cathedral#Abbey#House#Prison#Tower#Fort#Fair#Stadium#River#Bridge#Park#University#Theater

Table 7. Alternatives semantic information example

OntElectre procedure has been defined trying to reproduce the human thinking. With this objective, we designed a method that calculates the uncertainty corresponding to a criterion (i.e. indifference and preference thresholds) depending on the distance from the criterion class to the leaves, but also on the number of descendants of the class. Let us illustrate this with some examples. In the following figures the green square indicates the concepts values that belong to the corresponding alternative. The portion of the ontology corresponding to the criterion is shown in order to see the relation of the alternative values in each case: In the following, three criteria are considered:

- Suppose we were evaluating if alternative A is preferred to alternative B for criterion *Beach*:

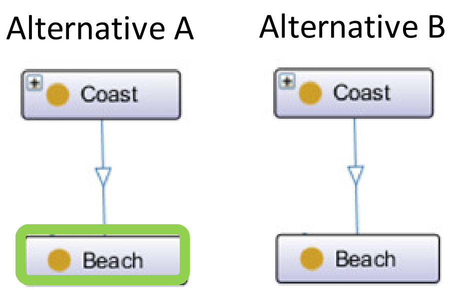


Figure 7. Management of the uncertainty example (I)

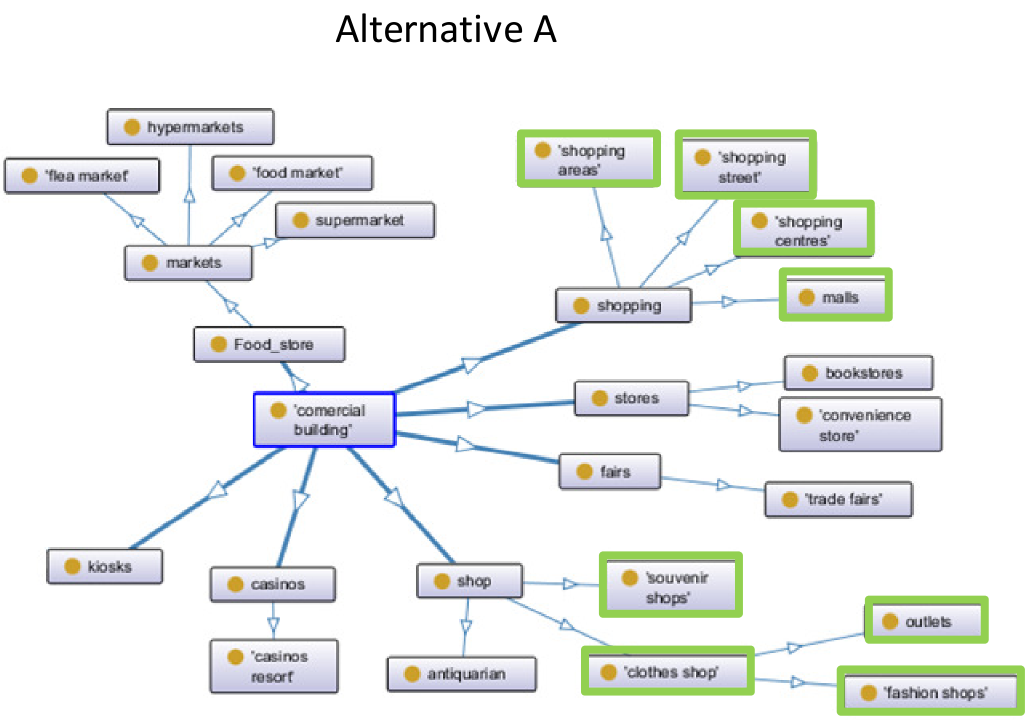
IF alternative A has the tag *Beach* and alternative B has not, there is no doubt: A is preferred than B.

- If we were evaluating criterion *Clothes shop*, the result would be equivalent:

Figure 8. Management of the uncertainty example (II)

Alternative A has two tags corresponding to *Clothes shop: Outlets* and *Fashion shops*. But, alternative B has only one tag corresponding to *Clothes shop: Outlets*. We surely would conclude that Alternative A is preferred to Alternative B.

- But what would happen if we were evaluating criterion *Commercial building:*



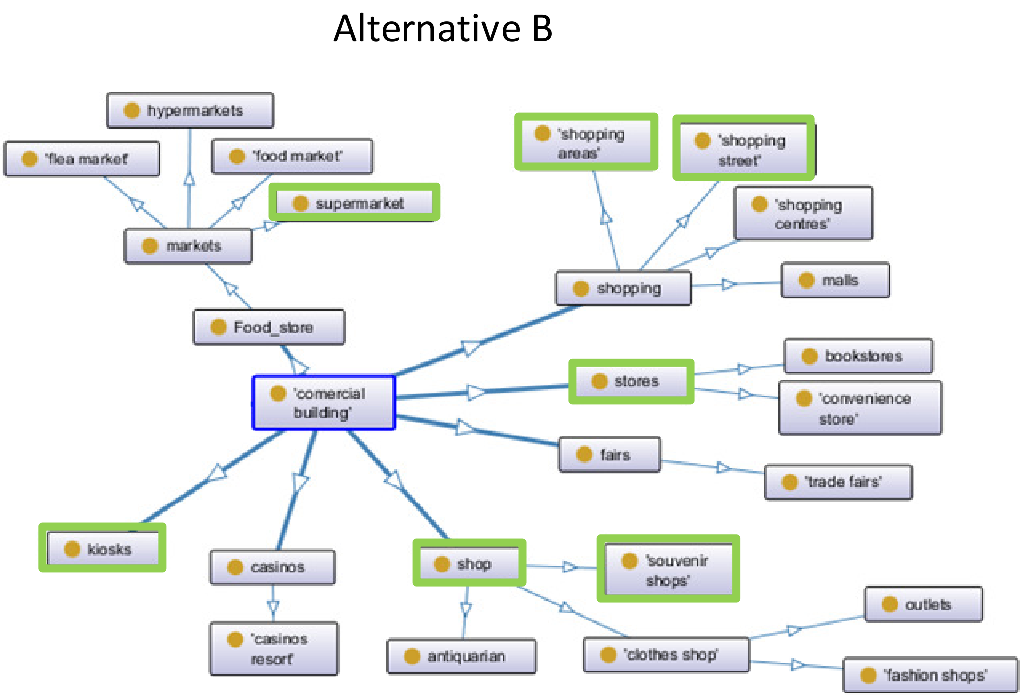


Figure 9. Management of the uncertainty example (III)

In this example, the decision would be much more difficult. Alternative A has one more tag than B, but this fact is not sufficient to conclude that A is preferred to B. A has more *Commercial building* tags, but all are concentrated in only two groups: *Shopping* and *Shop*, while B has more diversity of tags. Moreover, some of B tags have a small distance from *Commercial Building* than those from A. We would probably conclude that A and B are indifferent or incomparable alternatives.

With these simple examples, we have illustrated briefly our idea of comparing alternative with respect to semantic criteria, defined as subsets of a domain ontology. Following a pairwise comparison approach (as in ELECTRE outranking method), we have shown how two options could be compared to define a preference relation between them. As we have seen in the examples above, when the number of descendants of a class and/or the distance to leaves increase, the uncertainty also increases. We will reflect this in indifference and preference thresholds.

In following sections I will explain in detail the architecture and all the elements which form part of the proposed methodology.

5.1 Architecture

In Figure 10 we can see an overview of proposed methodology. Inputs of the system are the domain Ontology, the alternatives and the user profile. OntElectre generates xml files that will be Diviz input: alternatives, performance table, criteria and weights. We will use Diviz to run ELECTRE III with Distillation to generate the result ranking.

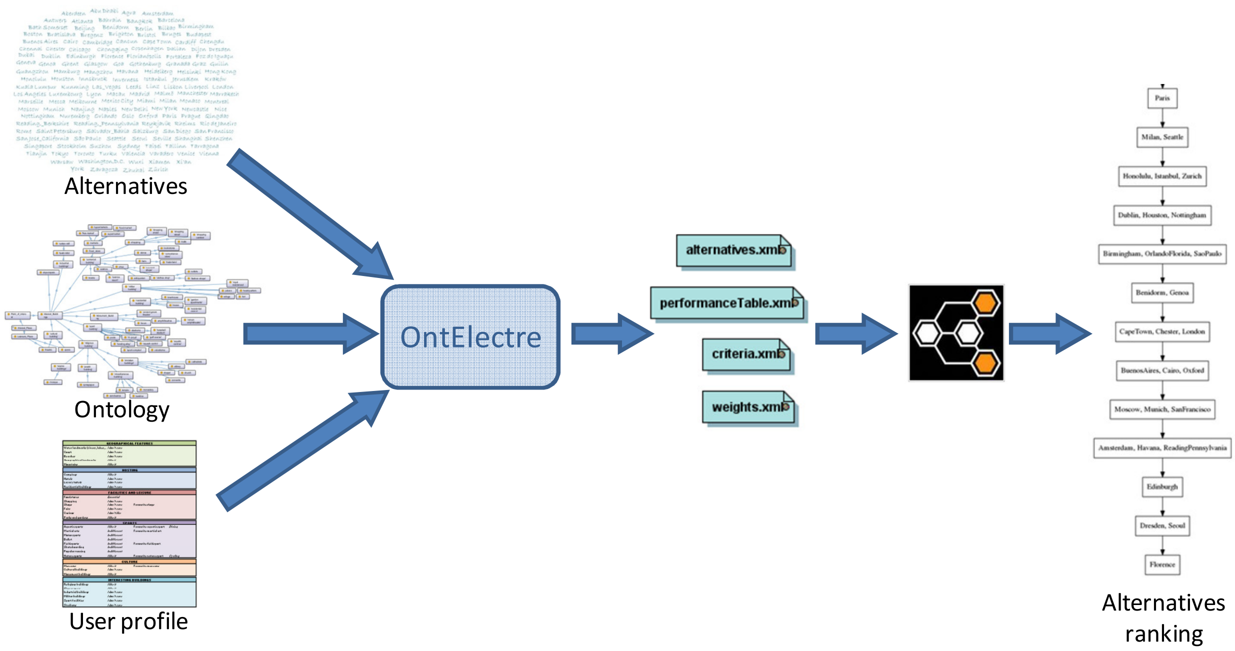


Figure 10. Proposed methodology architecture.

5.2 Ontology

As we have already explained in previous sections, the domain ontology is the base for semantically defining both the alternatives and the user profile. This domain ontology must correspond to an owl file in order to be managed by the system.

Taking into account that a domain ontology can be incredibly extensive and that thresholds will be proportional to the quantity of descendants of a class, a first step is to reduce the ontology, remaining only those classes whose descendants or the class itself appear in at least one of the alternatives lists of tags.

With this reduction of the ontology we achieve two important benefits:

1. We reduce the computing time. The system has to explore all the ontology topology in order to obtain the number of descendants and distance of each superclass. Having to explore unused classes supposes a waste of computing time.

2. ELECTRE thresholds will depend on the "existing" number of descendants of a class. If we maintained unused classes in the ontology, it could happen that obtained indifference threshold was higher than maximum number of used descendants of a class. In this case we would cancel the criterion because all alternatives would be indifferent for it.

   In Figure 11 we can see an example of this effect: Suppose we were calculating indifference threshold for criterion *A,* that corresponds to class *A* in the ontology. If we make an estimation assuming class *A* has sixteen descendants, a likely indifference threshold could be 2. But, in fact, in all the alternatives we could not find more than two values corresponding to *A* descendants (class *16* or class *8*), thus the maximum difference in performance values would be also 2. As a result all alternatives would be indifferent for criterion *A.* The maximum distance between leaves would also be incorrect if we do not remove he unused classes, as we would calculate a depth of 2 while the real maximum depth is 1.

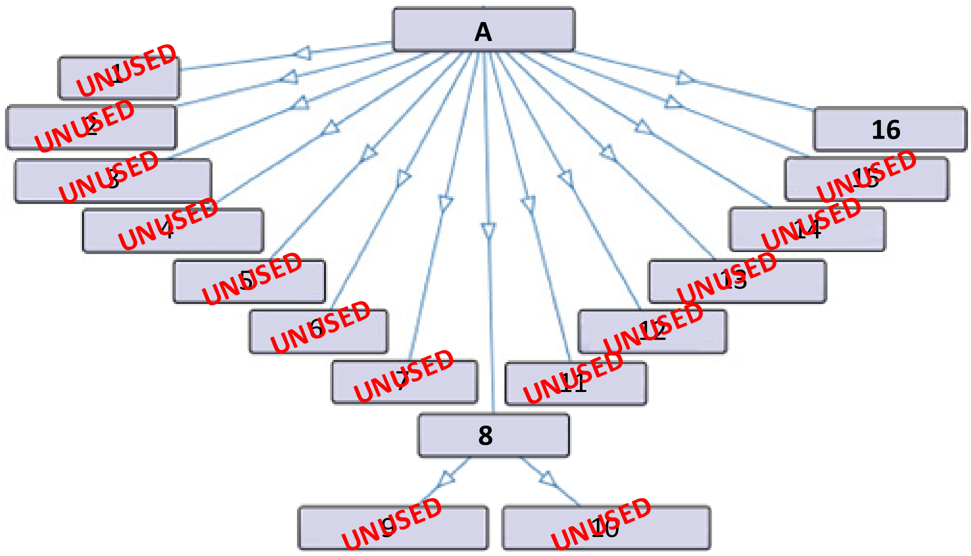


Figure 11. Unused classes effect

In order to implement the solution, we have extended Apache Jena OntModel class with additional methods, all of them used to estimate the thresholds or to calculate the performance table:

- `OntClass findClass(String className):` this method returns the OntClass object having className name.
- `boolean ontClassIsALeaf(OntClass pOntClass):` this method returns true if the class is a leaf (i.e. it has not descendants) and false otherwise.
- `int numOfLeaves(OntClass pOntClass):` this method returns the number of leaves descending from a class.
- `int numOfChildren(OntClass pOntClass):` this method returns the total number of descendants of a class. Descendants can be either leaves or superclasses (i.e. classes with descendants).
- `List<String> getChildren(OntClass pOntClass):` this method returns a list of all descendants of a class.
- `int getDistanceToLeaf(OntClass pOntClass):` this method returns the maximum distance from a class to their descendants being leaves.

5.3 Alternatives

Alternatives will be defined as follows:

- Alternative name
- Alternative tags = $\{a_1, a_2, ..., a_n\} \in$ Ontology classes names

Thus, every alternative has a name and a list of tags (corresponding to the features of this alternative). Tags of the alternatives must correspond directly with a class of the domain Ontology. There is no restriction in nature of the tags, they can correspond to every kind of class: superclasses or leaves.

Format of alternatives input file is a Excel file where the first column is the alternative name and second column contains the tags list separated by the symbol #.

5.4 User profile

The user profile contains the Decision Maker's preferences about semantic criteria. Each semantic criterion must correspond directly to a ontology class. Another fact that differentiates the proposed methodology with previous works is that in those the user profile only could contain preferences about leaves in the ontology. In OntElectre, by contrast, there is no restriction about the classes evaluated. The DM can express his/her preference about classes with descendants or about leaves.

The system is independent from the method to obtain the user profile. The user profile maintenance can be either implicit or explicit. User profile must contain a list of possible criteria and its preference value. In this first version of the system we will work with integer values of preference, this simplifies calculations and the maintenance of the user profile. Possible preference values are the following:

- **2 (Essential)**: it is mandatory that the ranked alternatives comply with this semantic feature. This is equivalent to having a performance > 0 for this criterion.

- **1 (The DM likes it)**: the DM likes this criterion, but it is not necessary. Alternatives with these features are preferred to others without.

- **-1 (The DM does not like it)**: the DM prefers alternatives that strictly do not comply with this feature. Alternatives with this feature will be considered worse than others without it.

- **0 (Indifferent or Unknown)**: this feature will not do the alternative better or worse. Simply the DM does not care about or we have no information. This feature will not be considered as a criterion and will not be evaluated.

Classes of the ontology with a preference value of 2, 1 or -1 will correspond to the criteria that will be evaluated for each alternative. Those with a preference value of 0 will not be considered as criteria and consequently will not be evaluated. Moreover, the possibility to let features marked as *Indifferent* makes the user profile management easier.

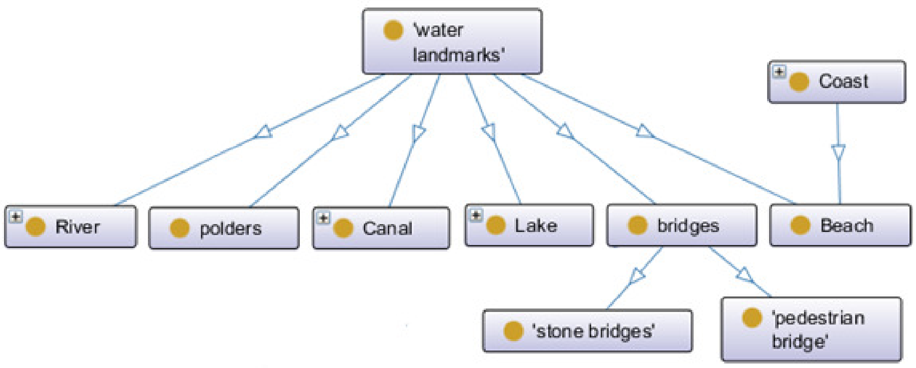
The behaviour of the system will be as follows depending on the preference value:

- **2 (Essential)**: alternatives that have a performance of zero for this criterion will be eliminated from the alternatives set. This helps to reduce ELECTRE computation time and it is useful for the DM as there is no sense in showing alternatives that do not comply with necessary preferences. Moreover, criterion marked as *essential* will have veto threshold.
- **1 (The DM likes it)**: criteria marked with *like* are included in the array of criteria and have to be maximized. These criteria will not have veto threshold.
- **-1 (The DM does not like it)**: criteria marked with *dislike* are included in the array of criteria and have to be minimized. These criteria will not have veto threshold.
- **0 (Indifferent)**: features marked as *indifferent* will not be either considered or evaluated as criteria. Apart from reducing the computation time, trying to infer a preference value for indifferent features could alter final results. Alternatives with more quantity of values would have always a higher punctuation even if they had not features of the DM interest.

5.5 Performance table

Performance table is a matrix with the performance value for each pair of alternative and criterion. The performance calculation is very simple: the system counts all the alternative tags that are descendants of the criterion or the criterion itself.

In Figure 12 we can see an example of performance calculation. Suppose we are calculating performance value for an alternative corresponding to the city of *Venice* and criterion *Water landmarks.* The system counts how many *Venice*'s features are descendants of class *Water landmarks*. We have to take into account that *Water landmark* could also be a *Venice*'s feature.



Venice:Sailing#Church#Cathedral#Monastery#Parish#Basilica#House#Palace#Tower#Fort#Art_Museum#Natural_History_Museum#Music_Museum#Biographical_Museum#Woman_Museum#**Canal**#**Beach**#**Lake**#**River**#**Bridge**#**Stone_Bridge**#Park#Statue#School#Opera#University#Library

Performance (Venice, Water landmarks) = 6

Figure 12. Performance calculation example

In order to calculate performances, the system maintains for each criterion a list of all its descendants. In this list we have to include the criterion itself.

5.6 Criteria and Thresholds

Criteria vector will be composed by all features in the user profile that have a preference value different from 0. This means the quantity of criteria is not fixed and it will depend on the user profile. Apart from the semantic criteria that have a direct match with the ontology topology, we could add to the criteria vector other numerical ones, like price or distance, which will not be related to the ontology. However in the case study we have only worked with semantic criteria in order to focus only on the semantic information without influences of other kind of criteria.

Each criterion is defined by following values:

- Criterion name
- Preference direction: possible values are *max* or *min*. This value indicates if the objective is to maximize or minimize the criterion depending on the preference value. With preference values > 0 the criterion wants to be maximized by the DM, with preference values < 0 the criterion wants to be minimized.
- Indifference threshold
- Preference threshold
- Veto threshold: if not informed it means the criterion has no veto.

As I have already mentioned in this work, the main key of the proposed methodology is in expressing the vagueness and the uncertainty using the discrimination thresholds. The more general the criterion is, the higher will be the values of these two thresholds. In following chapter I will explain in detail the thresholds calculation method.

5.6.1 Thresholds

As a reminder, I will describe the thresholds meaning as defined in the ELECTRE methodology:

- **Indifference threshold**: differences below this threshold are not big enough to consider an alternative is preferred to another.
- **Preference threshold**: if the difference in preference value is over the preference threshold one alternative will be considered preferred to the other. If the difference is between indifference and preference thresholds, concordance for this criterion will be calculated with an arithmetic function.
- **Veto threshold**: given two alternatives, if at least one criterion of the criteria array has a difference over the veto threshold, then the alternative with the worse preference for this criterion will automatically considered not preferred to the other.

Indifference and Preference thresholds will depend on two factors:

- The total number of descendants a class has.
- The distance from a class to its descendants.

Taking into account that preference values will be integer, we will do a round of thresholds obtained. Algorithm to calculate thresholds will be as follows:

if (criterion is a leaf) then
indifferenceThreshold = 0;
preferenceThreshold = 0;
else
*indifferenceThreshold =round(x*distanceToLeaves + y*numOfChildren);*
preferenceThreshold = round(z distanceToLeaves + k*numOfChildren);*
if (criterion is Essential) then
vetoThreshold = preferenceThreshold + w;
end if
end if

In order to determine *x*, *y*, *z*, *k* and *w* values we have to analyse both: the ontology topology and the alternatives features values. We can do empirical tests to deduce what are the best values.

There are some aspects to be considered when estimating *x*, *y*, *z*, *k* and *w* values:

- $q_i \leq p_i \leq v_i \;\; \forall\, i \; \in criteria$
  Veto threshold has to be equal or higher than preference threshold and preference threshold has to be equal or higher than indifference threshold for all criteria.

- $q_i < g_i(a) \quad \forall i \; \in criteria, \;\; \forall a \; \in alternatives$
  Indifference threshold obtained for a criterion can not be higher than the highest performance value it has in the Performance table. Suppose we are calculating indifference threshold for criterion *A* (q_A) and we estimate q_A = 10. Suppose maximum performance value for A is 5. In this case maximum difference in *A*'s performance we will obtain in all alternatives pairwise comparison will be 5, thus all alternatives will be indifferent in this criterion.

When calculating *numOfChildren* of a class we can do it from two different points of views:

- **Method 1** → *numOfChildren* = number of descendants in the ontology topology.
- **Method 2** → *numOfChildren* = maximum performance value in the alternatives set.

When analysing the difference between number of descendants in the ontology and maximum performance we can find two different situations:

- descendants in the ontology ≤ maximum performance → this happens when descendants of a class are more or less redundant. An example of this could be the class *Athletics* with descendants *running, jumping, throwing, walking, etc* . If a city had a stadium, probably its performance for *Athletics* would be the maximum and if the city had not, probably the performance would be zero.

- descendants in the ontology > maximum performance → this happens when descendants of the class are disjointed. An example of this could be the classification of a film. A film can be classified as *Romantic*, *Action, Historical*, *Fantastic*, *Drama, Children's film, Comical,* etc. The number of possible descendants would be high but rarely a film could have all the possible classifications at the same time.

Method 1 involves a better representation of the uncertainty but it can fall into the problem that indifference thresholds are higher than performance values and consequently we will cancel the criterion.

By the other hand, with Method 2 we will not fall into the problem of cancelling criteria, but we are loosing semantic information.

The method chosen will depend on the criteria and alternatives nature.

5.7 Weights

When we have to determine the weight of the different criterion we can classify them from two different perspectives:

- the liking degree the DM has expressed: essential or like.
- the nature of the criterion in the ontology definition: it has descendants or not.

From the perspective of the liking degree, obviously a criterion marked as essential by the DM must have a higher weight than another one that has only been marked as liked or disliked.

From the perspective of the nature of the criterion, we consider that a class with descendants may have a higher weight than a leaf (class without descendants). Let us see an example to illustrate it in Figure 13. The DM may have expressed liking in criterion *Ballet* and criterion *Field sport*. Having a good preference value in *Ballet* means that the alternative has the tag *Ballet*, there are only two alternatives: the city has it or not. But having a good preference value in *Field sport* means that the alternative has a high number of these tags: *Field sport, Golf, Volleyball, Table tennis, Badminton, Squash, Rugby, Cricket, Hockey, Ice hockey, Roller hockey, Tennis, Paddle, Basketball, Bowling, Handball* and *Football.* In consequence, we have defined that *Field sport* may have a higher weight than *Ballet.*

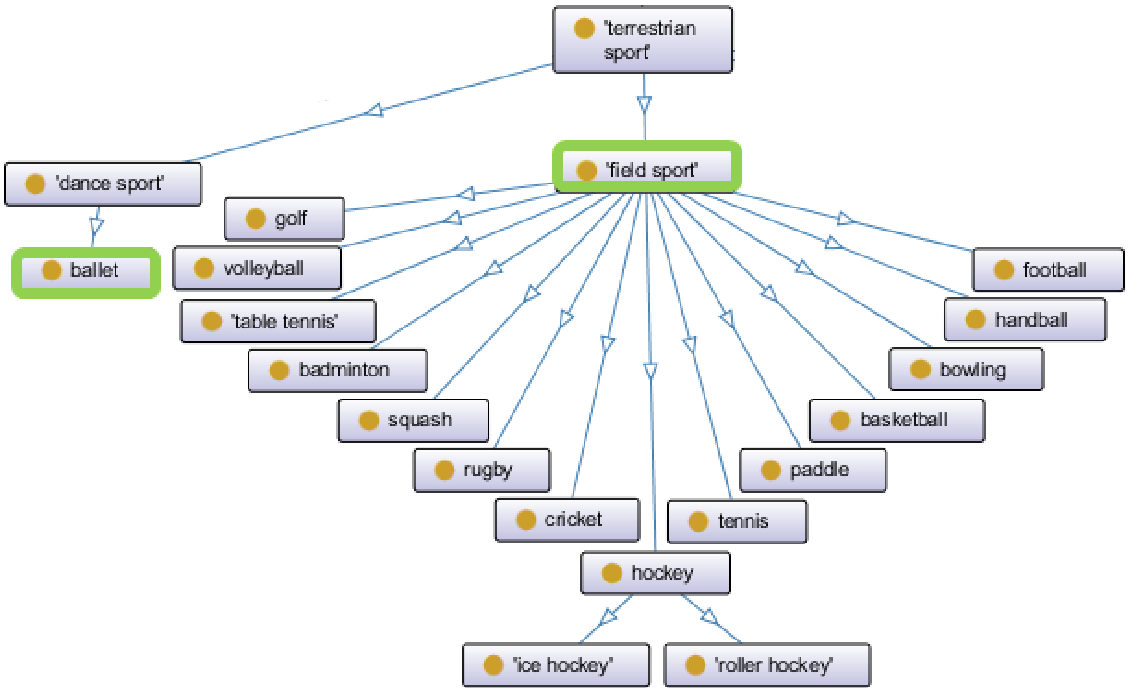


Figure 13. Different weights for leaves and superclasses.

In order to obtain the criteria weights, the system classifies them in four groups:

- **Essential leaf**: it is a criterion corresponding to a leaf (class without descendants) in the ontology that has been marked as essential by the DM. There is no sense in assigning a weight to these criteria because all alternatives in the set will have this tag with a performance of 1. This is due to the fact that the system removes alternatives that have a performance of zero in essential tags and leaves can have a maximum performance of one.
- **Leaf**: it is a criterion corresponding to a leaf in the ontology that has been marked by the DM as *liked* or *disliked*. We assume that having a good performance in a class with descendants has more weight than having only one tag corresponding to a leaf, consequently leaves will be counted once when obtaining the weights.
- **Superclass**: it is a criterion corresponding to a class with descendants in the ontology that has been marked by the DM as *liked* or *disliked*. Superclasses will be counted twice when obtaining the weights.
- **Essential superclass**: it is a criterion corresponding to a class with descendants in the ontology that has been marked by the DM as *Essential*. These will be the criteria with higher weight.

The procedure that we have defined to calculate the weights of the criteria is as follows:

1. First, we calculate the base to obtain the weights (x):

$$x = \frac{1}{numLeaves + (2 * numSuperclasses) + (3 * numEssentialSuperclasses)}$$

where:

- *numLeaves = number of criteria corresponding to a leaf marked as liked or disliked.*
- *numSuperclass = number of criteria corresponding to a class with descendants marked as liked or disliked.*
- *numEssentialSuperclass = number of criteria corresponding to a class with descendants marked as essential.*

2. Second, we calculate each criterion's weight following Table 8:

Evaluation	Is a leaf?	Weight
Essential	No	3x
Essential	Yes	0
I like it	No	2x
I like it	Yes	x
I don't like it	No	2x
I don't like it	Yes	x

Table 8. Weights calculation.

5.8 Diviz workflow

Diviz is a widely spread and useful tool for didactic purposes. It includes the implementation of most of the MCDA methods and the construction of new workflows is very simple. The user just does drag and drop of the selected module and connects outputs from a module to inputs of the next one. Another interesting Diviz feature is that user can analyse intermediate files generated by the modules. These are the principal reasons why I have chosen this tool to run ELECTRE III with Distillation.

In Figure 14 we can see the Diviz workflow used to generate the alternatives ranking. Workflow input files (alternatives.xml, performanceTable.xml, criteria.xml and weights.xml) are generated by OntElectre and are written in XMCDA standard.

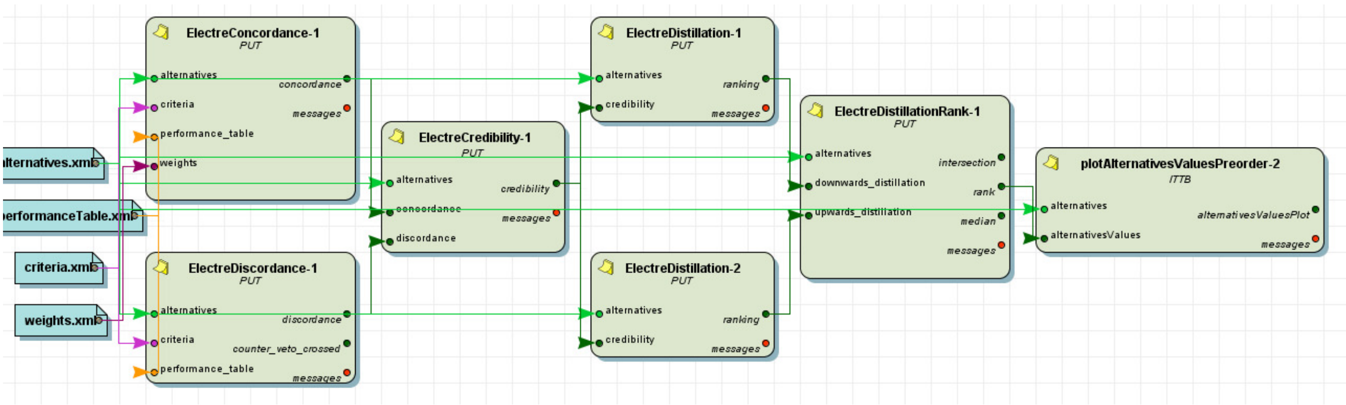


Figure 14. Diviz workflow

Following, I offer a brief explanation of each component:

- **alternatives.xml**: this file contains the alternatives description. Bellow an example of alternative definition is shown:

```
<alternative id="Aberdeen">
       <active>true</active>
</alternative>
```

- **performanceTable.xml**: this file contains the performance table (i.e. performance value for each alternative and each criterion). We can see an example bellow:

```
<performanceTable>
       <alternativePerformances>
              <alternativeID>Aberdeen</alternativeID>
              <performance>
                     <criterionID>Aquatic_Sport</criterionID>
                     <value><real>1.0</real></value>
              </performance>
              <performance>
                     <criterionID>Beach</criterionID>
                     <value><real>1.0</real></value>
              </performance>
              <performance>
                     <criterionID>Camping</criterionID>
                     <value><real>0.0</real></value>
              </performance>
       </alternativePerformances>
       ...
</performanceTable>
```

- **criteria.xml:** this file contains criteria definitions. For each criterion, we have to specify the criterion id and criterion name, the preference direction (the DM wants to maximize or minimize the criterion performance) and the criterion thresholds. When veto threshold is not informed it means that there is no veto for the criterion. In example bellow, we can observe that criterion *Aquatic_sport* has veto threshold while criterion *Beach* has no veto:

```
<criteria>
      <criterion id="Aquatic_Sport" name="Aquatic_Sport">
            <scale>
                  <quantitative>
                        <preferenceDirection>
                              max
                        </preferenceDirection>
                  </quantitative>
            </scale>
            <thresholds>
                  <threshold mcdaConcept="ind">
                        <constant><real>0.0</real></constant>
                  </threshold>
                  <threshold mcdaConcept="pref">
                        <constant><real>1.0</real></constant>
                  </threshold>
                  <threshold mcdaConcept="veto">
                        <constant><real>2.0</real></constant>
                  </threshold>
            </thresholds>
      </criterion>
      <criterion id="Beach" name="Beach">
            <scale>
                  <quantitative>
                        <preferenceDirection>
                              max
                        </preferenceDirection>
                  </quantitative>
            </scale>
            <thresholds>
                  <threshold mcdaConcept="ind">
                        <constant><real>0.0</real></constant>
                  </threshold>
                  <threshold mcdaConcept="pref">
                        <constant><real>0.0</real></constant>
                  </threshold>
            </thresholds>
      </criterion>
      ...
</criteria>
```

- **weights.xml**: this file contains criteria weights. Bellow, an example is shown:

```
<criteriaValues mcdaConcept="Importance" name="significance">
      <criterionValue>
            <criterionID>Aquatic_Sport</criterionID>
            <value><real>0.15000000000000002</real></value>
      </criterionValue>
      <criterionValue>
            <criterionID>Beach</criterionID>
            <value><real>0.0</real></value>
      </criterionValue>
      ...
</criteriaValues>
```

- **ElectreConcordance**: this module calculates the overall concordance matrix. This matrix contains the comparison of all alternatives pairs.

Alternatives comparisons

- Barcelona R Barcelona (1.0)
- Barcelona R BathSomerset (1.0)
- Barcelona R Birmingham (1.0)
- Barcelona R Bruges (1.0)
- Barcelona R BuenosAires (1.0)
- Barcelona R Chester (1.0)
- Barcelona R Copenhagen (0.846153846154)
- Barcelona R Hamburg (1.0)
- Barcelona R Heidelberg (0.846153846154)
- Barcelona R Krakow (0.846153846154)
- Barcelona R Kunming (1.0)

- **ElectreDiscordance**: this module computes the discordance matrix. This matrix consists in the discordance between all alternatives pairs. When there is at least a 1 value in the discordance vector, this means that the first alternative has reached the veto threshold (i.e. it does not outrank the second one). This module generates a Cartesian product between all alternatives with all alternatives and all criteria. In proposed solution, when considering all alternatives (149) for all possible criteria (38), it generated a file of 149x149x38 = 843638 elements that was almost impossible to handle.

Alternatives comparisons

- Barcelona R Barcelona (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R BathSomerset (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R Birmingham (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R Bruges (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R BuenosAires (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R Chester (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R Copenhagen (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)
- Barcelona R Hamburg (
  Archeology_Museum:0.0 Cultural_Building:0.0 Militar_Building:0.0 Monument_Building:0.0 Museum:0.0 Religious_Building:0.0)

- **ElectreCredibility**: this module computes credibility matrix using procedure which is common to the most methods for the ELECTRE family.

Alternatives comparisons

- Barcelona R Barcelona (1.0)
- Barcelona R BathSomerset (1.0)
- Barcelona R Birmingham (1.0)
- Barcelona R Bruges (1.0)
- Barcelona R BuenosAires (1.0)
- Barcelona R Chester (1.0)
- Barcelona R Copenhagen (0.846153846154)
- Barcelona R Hamburg (1.0)
- Barcelona R Heidelberg (0.846153846154)
- Barcelona R Krakow (0.846153846154)
- Barcelona R Kunming (1.0)
- Barcelona R Lisbon (0.846153846154)

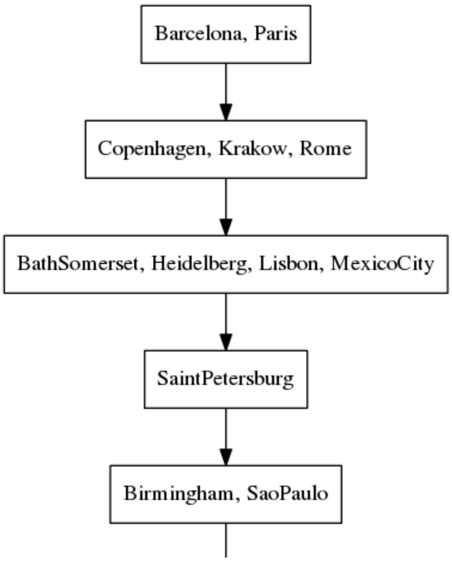
- **ElectreDistillation**: this module performs the distillation process. We have to include in the flowchart two ElectreDistillation modules one performs upwards distillation and the other downwards distillation.

- **ElectreDistillationRank**: this module performs the intersection between upwards and downwards distillations. With this module we obtain the alternatives rank, best alternatives have the smallest value:

Alternatives values

Barcelona	1
BathSomerset	3
Birmingham	5
Bruges	8
BuenosAires	6
Chester	6
Copenhagen	2
Hamburg	7
Heidelberg	3
Krakow	2
Kunming	10
Lisbon	3
MexicoCity	3
Milan	8

- **plotAlternativesValuesPreorder**: this module generates a graph representing a ranking on the alternatives, according to results obtained in previous module.



5.9 Summary and discussion

In this section I have described the proposed methodology that uses semantic techniques based on ontologies to generate a ranking of the alternatives by means of the ELECTRE methodology. Main characteristics of this solution are the following:

- System's inputs are: a domain ontology, an alternatives' file and the user profile.
- Alternatives are described by tags that correspond to ontology's classes.
- User profile contains the DM preferences about criteria that correspond to ontology's classes.
- Criteria's preferences are expressed with integer values: 0 (Indifferent), 1 (Liked), 2 (Essential), -1 (Disliked).
- The number of semantic criteria is flexible. The system does not handle criteria with unknown preference value or that are indifferent for the DM.
- Other numerical criteria can be added to the criteria array.
- Performance of a criterion for an alternative is calculated as the intersection between criterion's descendants in the ontology and alternative's tags list.
- Uncertainty is represented by thresholds. Thresholds are proportional to the number of descendants of a class and to the distance to leaves.
- Criteria's weights depend on the criterion's preference value and on the nature of the class: leaf or superclass.
- A standard Outranking MCDA method can be used to generate the alternatives ranking. In the proposed methodology ELECTRE III with distillation.

5.9.1 Comparison with previous related works

Table 9 summarizes the comparison of the features between the presented proposed methodology (OntElectre) and those designed in previous works for semantic criteria, being [Bor12][Mar16]:

Feature	Bor12	Mar16	OntElectre
It is a semantic multi-criteria method	No, all semantic preferences are aggregated in only one numerical criterion.	Each ontology defines a unique criterion, which is linguistic and multivalued	Yes, we can define several criteria on a single ontology.
Estimates unknown values	Yes, with a upwards and downwards propagation	Yes, estimates unknown values using the taxonomical relationships on the ontology	No, unknown values are ignored
Can cause compensation	Yes, because aggregation is calculated with a modification of OWA aggregation	No, because the concepts are compared in a pairwise procedure using the Semantic Win Rate.	No, criteria are treated separately
Handles uncertainty	Yes, with the Confidence Level	Yes, with a new definition of the concordance and discordance fuzzy functions, defining a new ELECTRE approach	Yes, calculating the best values for the thresholds of standard ELECTRE.
Uses a standard outranking MCDA method	Yes, ELECTRE ranking exploitation	No, it defines an enhancement of ELECTRE-III	Yes, ELECTRE III explotiation
Enables expressing superclasses' preference	No, user preferences are only expressed in leaves	No, user preferences are only expressed in leaves	Yes

Table 9. Comparison with related works

We can see that the new proposal, OntElectre, has different features in comparison to the previous ones. Therefore, it constitutes a new approach to the management of semantic data in recommendation processes based on an outranking methodology.

6 Case study

I have presented in this Master Thesis the design of a new methodology that enables applying the standard Outranking MCDA method ELECTRE to a set of alternatives described by semantic multi-valued attributes and with a user profile based in a domain ontology.

As a case study, this system will be applied to the Tourism domain, concretely to the recommendation of touristic city destinations.

The data used in the case study comes from the DAMASK (Data Mining Algorithms with Semantic Knowledge) research project, conducted at the ITAKA research group at URV. The DAMASK project proposes the use of semantic domain knowledge, represented in the form of ontologies in different tools that are needed to develop Recommender Systems on the Web. The project is conformed in the strategic area of Tourism. In this project, a Web application for the personalised classification of touristic destinations has been developed and studied. The main contributions of this previous work in DAMASK are related to two issues: first, the collection of structured data from the textual information available in the Web in an automatic way and, second, the use of ontologies for the management and analysis of the collected semantic data. However, the analytical approach was different from the current one because it was based on the use of clustering algorithms and semantic similarity measures.

In this master thesis, two resources created in the DAMASK project will be used for the testing of the proposed new methodology: DAMASK tourism ontology and DAMASK matrix of the most touristic cities in the world.

6.1 DAMASK Ontology

DAMASK ontology is the result of merging and combining different ontologies related in the area of tourism: **TourismOWL.owl**, **Space.owl** and **PCTTO.owl**. It consists of 538 classes connected in 9 hierarchy levels. It is structured around 4 main concepts that constitute the first level of the hierarchy: “geopolitical division”, “activity”, “point of interest” and “geographical feature”. The Damask Ontology is not a pure taxonomy, as it contains multi-inheritance between concepts [Mor11][Vic11].

As explained in section 5.2 a reduction of the original domain ontology is needed. I did this reduction manually, removing from the owl file all classes that did not appear in the alternatives matrix and were not ancestors of other classes that appeared in the alternatives matrix.

In Annexes section some images of this reduced version of DAMASK Ontology are presented.

6.2 Alternatives

I will use the DAMASK City Matrix as alternatives set. This matrix was constructed in the Master Thesis of Carlos Vicient[Vic10] and completed later by Ferran Mata [Mat12]. First step was the construction of a data matrix that comprehends the cities and their description using heterogeneous attributes. This data matrix was built using the data extracted in previous steps of the DAMASK project. Sources to obtain information were heterogeneous Web resources like Wikipedia and specific APIs of different Websites like Geonames and Google Maps Elevation.

In previous steps of the DAMASK project, Carlos Vicient [Vic09][Vic10] applied methods of semantic extraction and integration of information that were used to obtain a data matrix that gathers all the information available in Web resources about some touristic destinations. The tools developed in this previous task of the project were used by Mata to complete the matrix.

A set of 150 cities distributed all over the world was selected. These cities were the 150 leading and most dynamic cities in terms of tourist arrivals, according to the ranking made by Euromonitor International in 2006 [Bre07].

The cities are the following: Aberdeen, Abu Dhabi, Agra, Amsterdam, Antwerp, Atlanta, Bahrain, Bangkok, Barcelona, Bath (Somerset), Beijing, Benidorm, Berlin, Bilbao, Birmingham, Boston, Bratislava, Bregenz, Brighton, Bristol, Bruges, Budapest, Buenos Aires, Cairo, Cambridge, Cancun, Cape Town, Cardiff, Chengdu, Chennai, Chester, Chicago, Chongqing, Copenhagen, Dalian, Dijon, Dresden, Dubai, Dublin, Edinburgh, Florence, Florianopolis, Fortaleza, Foz do Iguaçu, Geneva, Genoa, Ghent, Glasgow, Goa, Gothenburg, Granada, Graz, Guangzhou, Guilin, Hamburg, Hangzhou, Havana, Heidelberg, Helsinki, Hong Kong, Honolulu, Houston, Innsbruck, Inverness, Istanbul, Jerusalem, Krakow, Kuala Lumpur, Kunming, Las Vegas (Nevada), Leeds, Linz, Lisbon, Liverpool, London, Los Angeles, Luxembourg, Lyon, Macau, Madrid, Malmö, Manchester, Marrakech, Marseille, Mecca, Melbourne, Mexico City, Miami, Milan, Monaco, Montreal, Moscow, Mumbai, Munich, Nanjing, Naples, New Delhi, New York City, Newcastle upon Tyne, Nice, Nottingham, Nuremberg, Orlando (Florida), Oslo, Oxford, Paris, Prague, Qingdao, Reading (Berkshire), Reading (Pennsylvania), Reykjavík, Rheims, Rio de Janeiro, Rome, Saint Petersburg, Salvador (Bahia), Salzburg, San Diego, San Francisco, San Jose (California), São Paulo, Seattle, Seoul, Seville, Shanghai, Shenzhen, Singapore, Stockholm, Suzhou, Sydney, Taipei, Tallinn, Tarragona, Tianjin, Tokyo, Toronto, Turku, Valencia, Spain, Varadero, Venice, Vienna, Warsaw, Washington D.C., Wuxi, Xiamen, Xi'an, York, Zaragoza, Zhuhai, Zürich.

Damask data matrix contains two different types of information for each city:

- **Categorical and numerical data**: Population, Elevation, Continent code and Climate.
- **Semantic data**: consists in a set of semantic variables (tags) whose values correspond to DAMASK ontology classes. These values in the variables can be either a leaf (class without descendants) in the ontology or a superclass (class with descendants). In some cases, an alternative has a tag corresponding to a superclass and another one corresponding to a descendant of it.

In original Data Matrix, semantic variables were organized in different groups: Aquatic + Nature Sports, Other sports, Religious Building, Other Buildings, Museum, Landmark, Other Landmark and Cultural Building.

	A	H	I	J	K	L	M
1	City-name	Other Landmark (Park + Column + Conmemorate + Memorial + Tomb)	Cultural_Building	Population	Elevation	Continent code	Climate
2	Aberdeen	Park#Fountain	University#Theater#Private_School	183790	10.45	EU	Oceanic
3	Abu_Dhabi	Park	School#Library	603492	6.03	AS	Desert
4	Agra	Statue#Mausoleum#Tomb	Public_University#Public_School#University#School	1430055	165.22	AS	Semi-arid
5	Amsterdam	Nature_Reserve#Zoo	Theater#School#Opera#University	741636	12.66	EU	Oceanic
6	Antwerp	Zoo#Statue#Tomb	Theater#School#University	459805	12.44	EU	Oceanic
7	Atlanta	Botanical_Garden#Zoo#Park#Statue	Public_University#Theater#Public_School#Art_School#Opera#University#School	420003	319.92	NA	Humid sub-tropical
8	Bahrain	Fountain	University	738004	4.96	AS	Desert
9	Bangkok	Green_Zone#Forest_Park#Park#Statue	Theater#School#University	5104476	11.92	AS	Tropical savanna
10	Barcelona	Historic_Park#Zoo#Urban_Park#Forest_Park#Park#Statue#Crypt	Theater#Opera#Forum#University#Ancient_Greek_Theatre	1621537	30.70	EU	Mediterranean
11	Bath,_Somerset	Botanical_Garden#Park#Ionic_Column#Column#Statue	University#Theater#School#Forum#Amphitheatre#Opera#Library	93238	22.61	EU	Oceanic
12	Beijing	Botanical_Garden#Zoo#Garden_Park#Obelisk#Mausoleum	School#Opera#University#Library	7480601	50.09	AS	Humid continental
13	Benidorm	Park#Zoo	?	71034	27.67	EU	Mediterranean
14	Berlin	Zoo#Botanical_Garden#Column#Fountain#Crypt	University#School#Opera#Theater#Library	3426354	43.90	EU	Oceanic
15	Bilbao	Park	University#Theater#Opera	354860	14.79	EU	Oceanic
16	Birmingham	Botanical_Garden#Nature_Reserve	University#Theater#School#Library#Forum#Music_School#Opera	984333	153.14	EU	Oceanic
17	Boston	Zoo#Park	Public_University#Theater#Public_School#Opera#Library#Technological_University#Music_School#University	617594	15.28	NA	Humid sub-tropical
18	Bratislava	Botanical_Garden#Zoo#Forest_Park#Statue#Pyramid	University#Theater#Opera#Library#Business_School	423737	156.15	EU	Oceanic
19	Bregenz	?	Theater#Opera	27611	399.65	EU	Oceanic
20	Brighton	Park#Mausoleum	University#Theater#School	139001	26.49	EU	Oceanic
21	Bristol	Zoo#Nature_Reserve#Park#Statue#Megalithic	University#Theater#School	430713	22.48	EU	Oceanic
22	Bruges	Park#Statue	Theater#School#Opera#Forum#University	116709	13.00	EU	Oceanic
23	Budapest	Park#Statue#Tomb	University#Opera#Library#Theater	1696128	109.18	EU	Oceanic
24	Buenos_Aires	Botanical_Garden#Zoo#Park#Obelisk#Statue	University#Theater#School#Opera#Library#Ancient_Greek_Theatre#Business_School	13076300	31.83	SA	Humid sub-tropical
25	Cairo	Park#Pyramid	University#Theater#Opera#Library#School	7734614	24.21	AF	Desert
26	Cambridge	Park#Zoo#Fountain	University#Theater#School#Library#Opera	128488	11.67	EU	Oceanic
27	Cancún	Park	?	542043	10.01	NA	Tropical savanna
28	Cape_Town	Park#Statue	Theater	3433441	26.19	AF	Mediterranean
29	Cardiff	Nature_Reserve#Park#Pyramid#Megalithic	University#Theater#Opera#Library#School	302139	16.78	EU	Oceanic

Figure 15. Original DAMASK Cities Matrix

Taking into account that the main goal of OntElectre is to determine a system to generate a ranking of touristic destinations starting from an ontology profile, I will only use semantic columns of this matrix.

I have also formatted the file unifying all data in the same column and removing the "?" symbols (i.e. missing values). When removing the "?" and the categorical and numerical tags I found Mumbai had no information in any variable, so I removed this city from the alternatives set.

Finally, the set of alternatives is composed by 149 cities, as shown in Figure 16.

Figure 16. Alternatives set.

6.3 Acquisition of the User profile

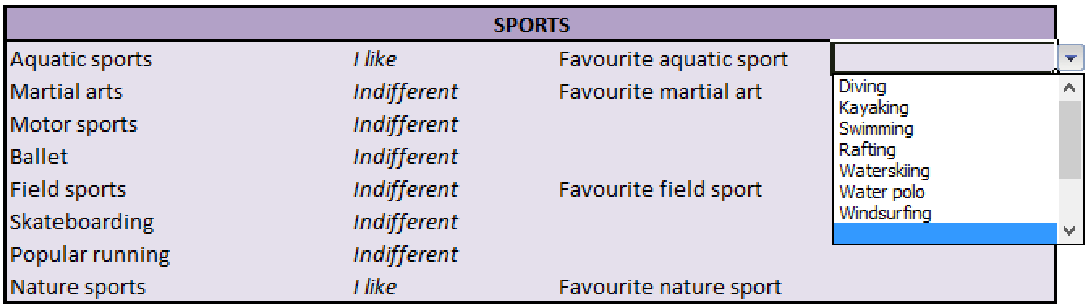
6.3.1 Description

In this case study, the method to obtain the user profile will be explicit. The Decision Maker (DM) has to answer a simple interview in which there are only four possible answers for each criterion evaluated:

- **Essential**: the DM is looking for a touristic destination that must comply with this feature.
- **I like it**: the DM likes destinations that comply with this feature, but are not necessary. Alternatives with these features are preferred to others without.
- **I don't like it**: the DM prefers destinations that strictly do not comply with this feature. Cities with this feature will be considered worse than others without this feature. It is important that the DM understands the difference between *Indifferent* and *I don't like it*. For instance, I like Parks and I do not like Casinos, but for me, a city with a park and a casino will not be considered worse than another one that has a park and does not have a casino. In this example I should mark casinos as indifferent.
- **Indifferent**: this feature will not do the alternative better or worse. Simply the DM does not care about this. This feature will not be considered as a criterion and will not be evaluated.

A support questionnaire has been implemented in Excel Each question matches directly with a class in DAMASK ontology. It can be a class with descendants or a leaf (i.e. a class without descendants). The possibility to let features marked as *Indifferent* makes it easier for the DM to answer the questionnaire.

In some cases, when the criterion evaluated is a class with descendants, it is also possible to choose a favourite subclass of the evaluated classes. In Figure 17 we can see an example of the interview sheet that has been implemented. When the Decision Maker likes *Aquatic sports* he/she can choose which aquatic sport prefers.



SPORTS			
Aquatic sports	*I like*	Favourite aquatic sport	
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*Indifferent*		
Field sports	*Indifferent*	Favourite field sport	
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*I like*	Favourite nature sport	

Figure 17. Example of choosing a favourite aquatic sport.

In other cases, in the same section it is possible to choose a superclass and a leaf class descendant of it. This is equivalent to adding an "extra point" to alternatives that complain with both classes, the most generalist and the concrete one. For instance, in Figure 18 we can see that in the block Geographical features the DM can say that likes *Geographical landmarks* and *Mountains*. Cities with mountains will increase the performance of criterion *Geographical landmarks* and, at the same time, obtain an "extra point" in criterion *Mountains.*

GEOGRAPHICAL FEATURES	
Water landmarks (rivers, lakes,...)	*I don't care*
Coast	*I don't care*
Beaches	*I don't care*
Geographical landmarks	*I like it*
Mountains	*I like it*

Figure 18. Example of choosing a superclass and a descendant

The intention was to elaborate a questionnaire easy to answer, for this reason, principal questions are more general while favourite questions let the DM to be more specific, when preferred.

For the elaboration of the interview I have selected a subset of key ontology classes that can be considered of touristic interest and they will constitute the set of criteria of this case study. In the selection, I have mixed classes in different levels in order to test exhaustively the system.

For each criterion a question is made o the user in order to determine his/her preference. Criterias are organised in six blocks: *Geographical features, Hosting, Facilities and leisure, Sports, Culture* and *Interest Buildings.* They are detailed in the following subsections, showing in green the selected classes of the different ontology parts.

6.3.2 User profile: Geographical features

In this block the DM can answer his/her preferences about the geographical features of the city. Classes selected to be evaluated are three superclasses: *Water landmarks*, *Coast* and *Geographical landmarks* and two leaves: *Mountains* and *Beaches.*

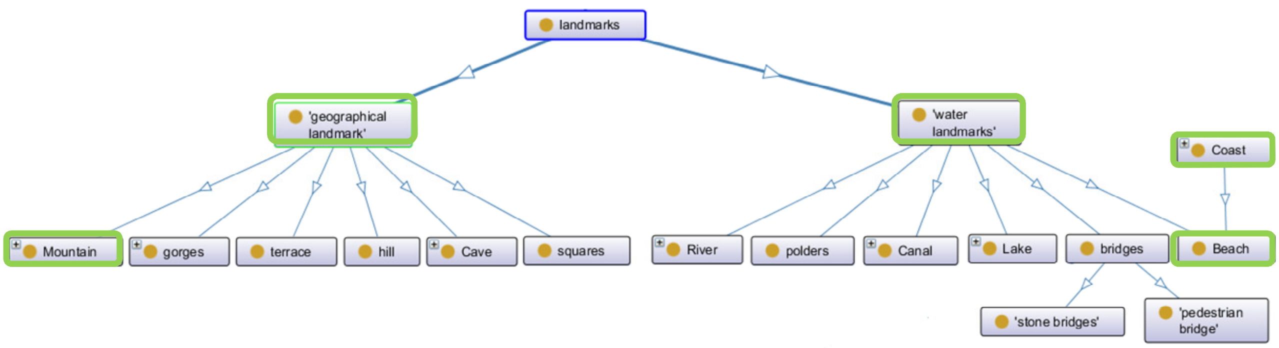


Figure 19. Block Geographical features ontology graph.

6.3.3 User profile: Hosting

In this block the DM may answer about his accommodation preferences. These can be: *Camping, Hotel, Luxury hotel* or *Residential building. Residential building* and *Hotel* are superclasses while *Camping* and *Luxury hotel* are leaves.

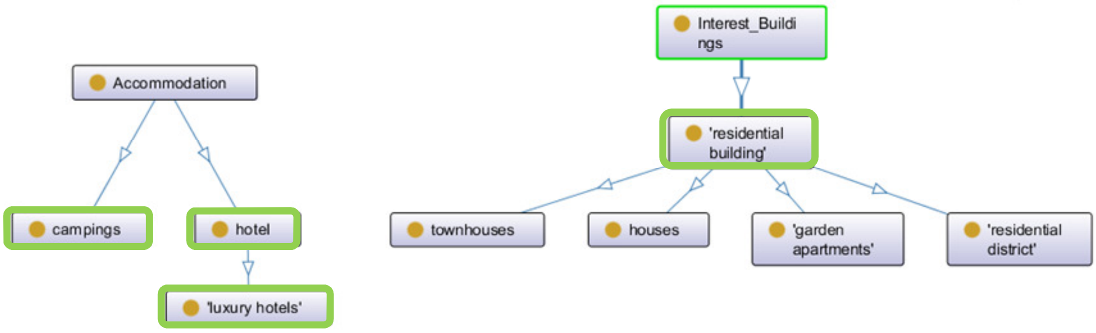


Figure 20. Block Hosting Ontology graph.

6.3.4 User profile: Facilities and leisure

In this block, the DM answers principally questions about shopping: *Fair, Food store, Shop* and *Shopping* and other questions about leisure: *Casino* and *Parks*. All criteria evaluated in this block are superclasses.

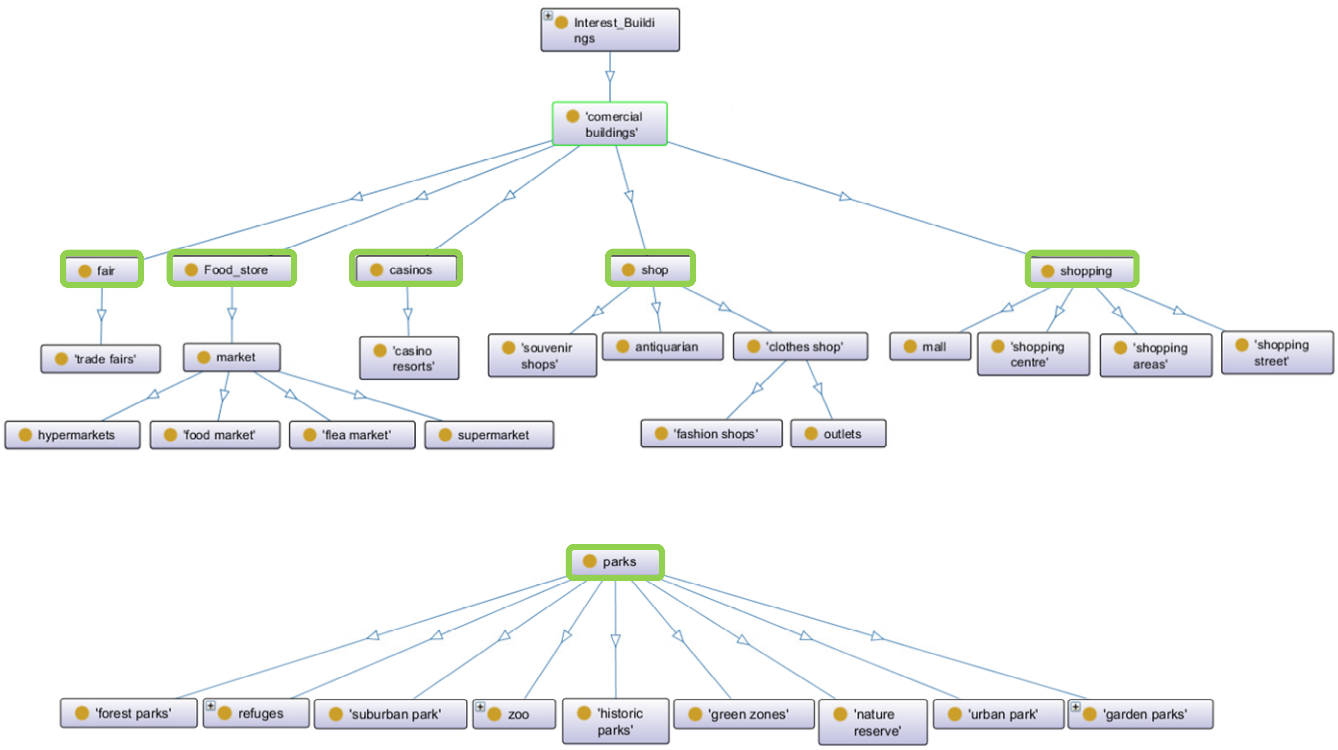


Figure 21. Block Facilities and leisure Ontology graph.

6.3.5 User profile: Sports

In this block, the DM can answer about his sports preferences. Sports classes, especially field sports, have a big number of leaves. For this reason, the DM can answer about superclasses preferences and after, for *Aquatic sport, Martial art, Field sport* and *Nature sport* he can choose one of the favourites.

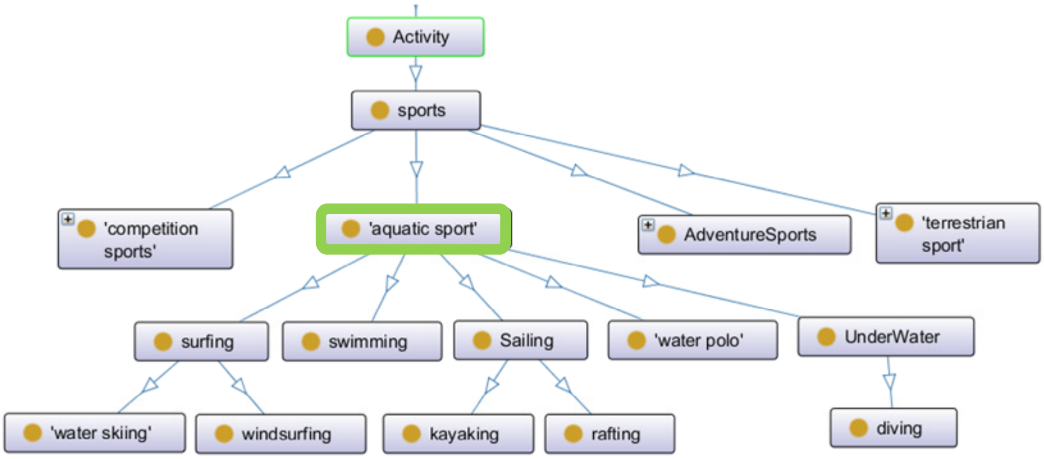


Figure 22. Block sport Ontology graph (I)

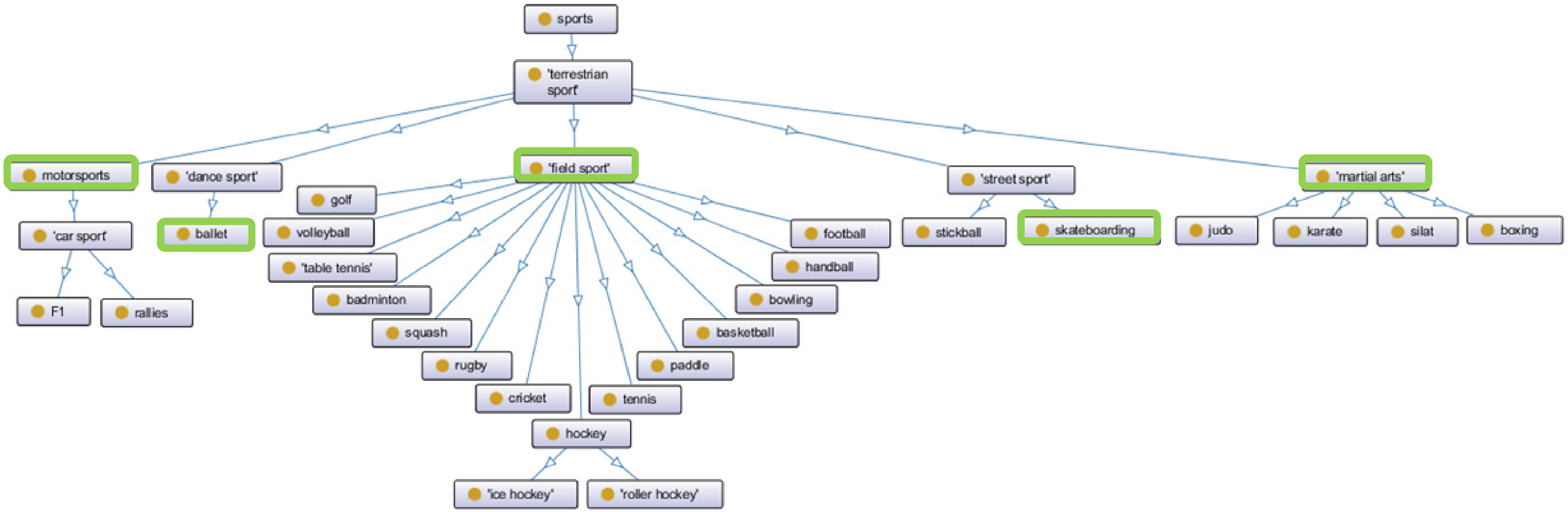


Figure 23. Block sport Ontology graph (II)

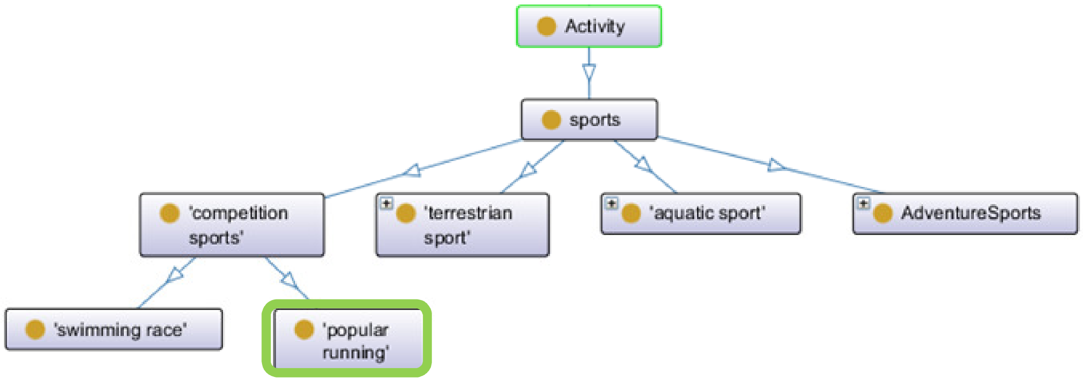


Figure 24. Block sport Ontology graph (III)

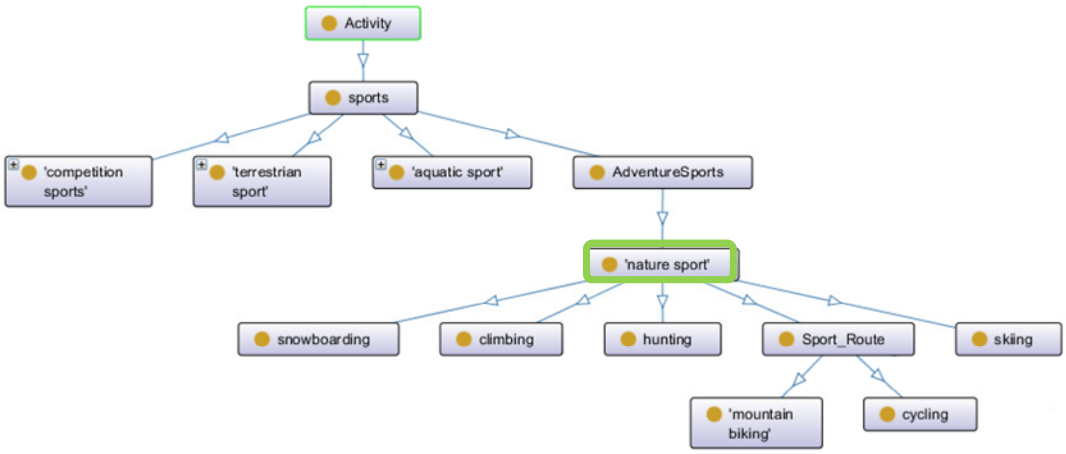


Figure 25. Block sport Ontology graph (IV)

6.3.6 User profile: Culture

In this block I have mixed classes descendant from different branches that can be considered cultural features: *Museums*, *Cultural buildings* and *Monument buildings.*

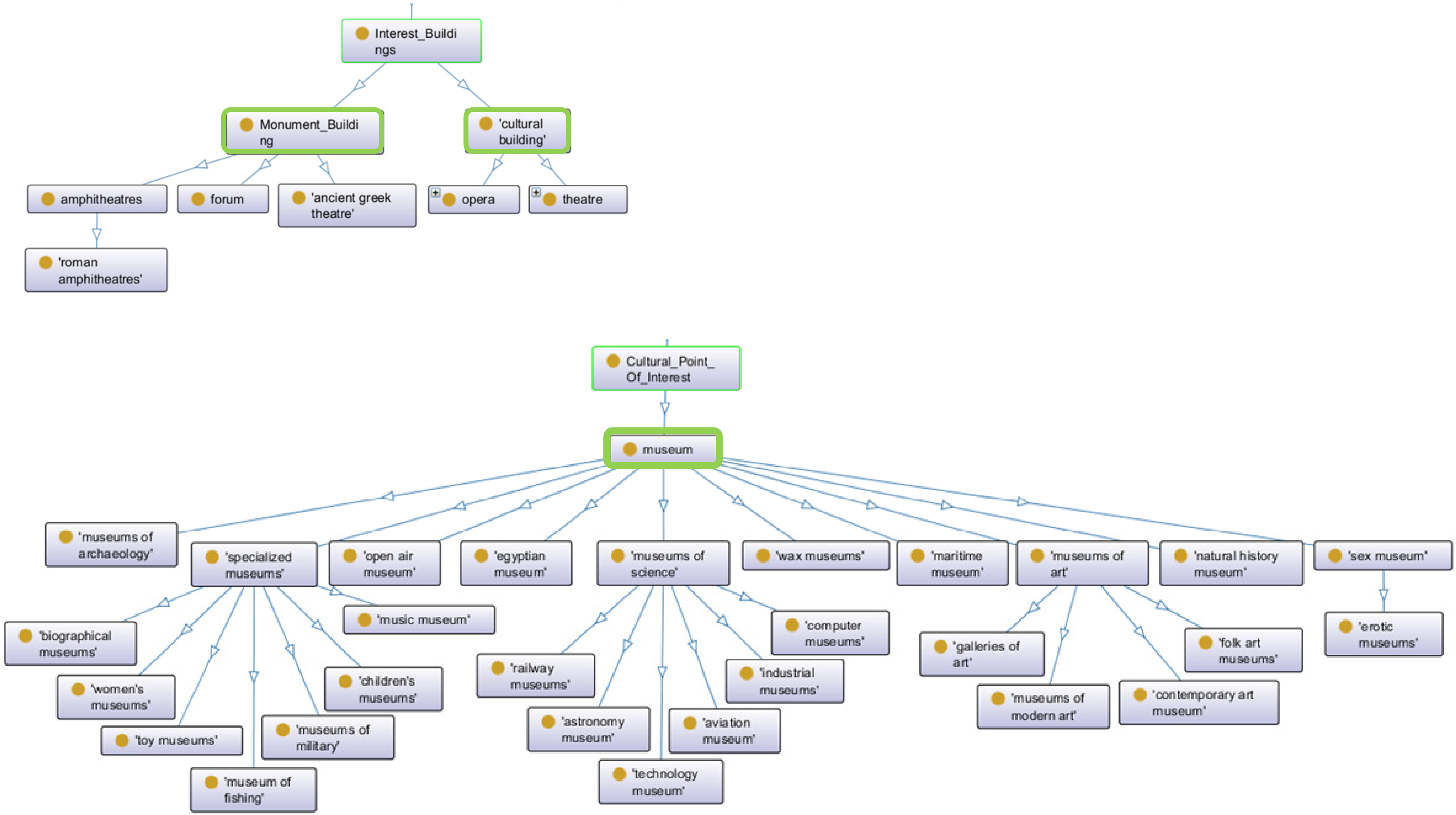


Figure 26. Block culture Ontology graph

6.3.7 User profile: Interest Buildings

In this block the DM can answer about his preferences about interest buildings: *Religious buildings, Military buildings, Industrial buildings, Skyscrapers, Sport buildings* and *Stadiums.*

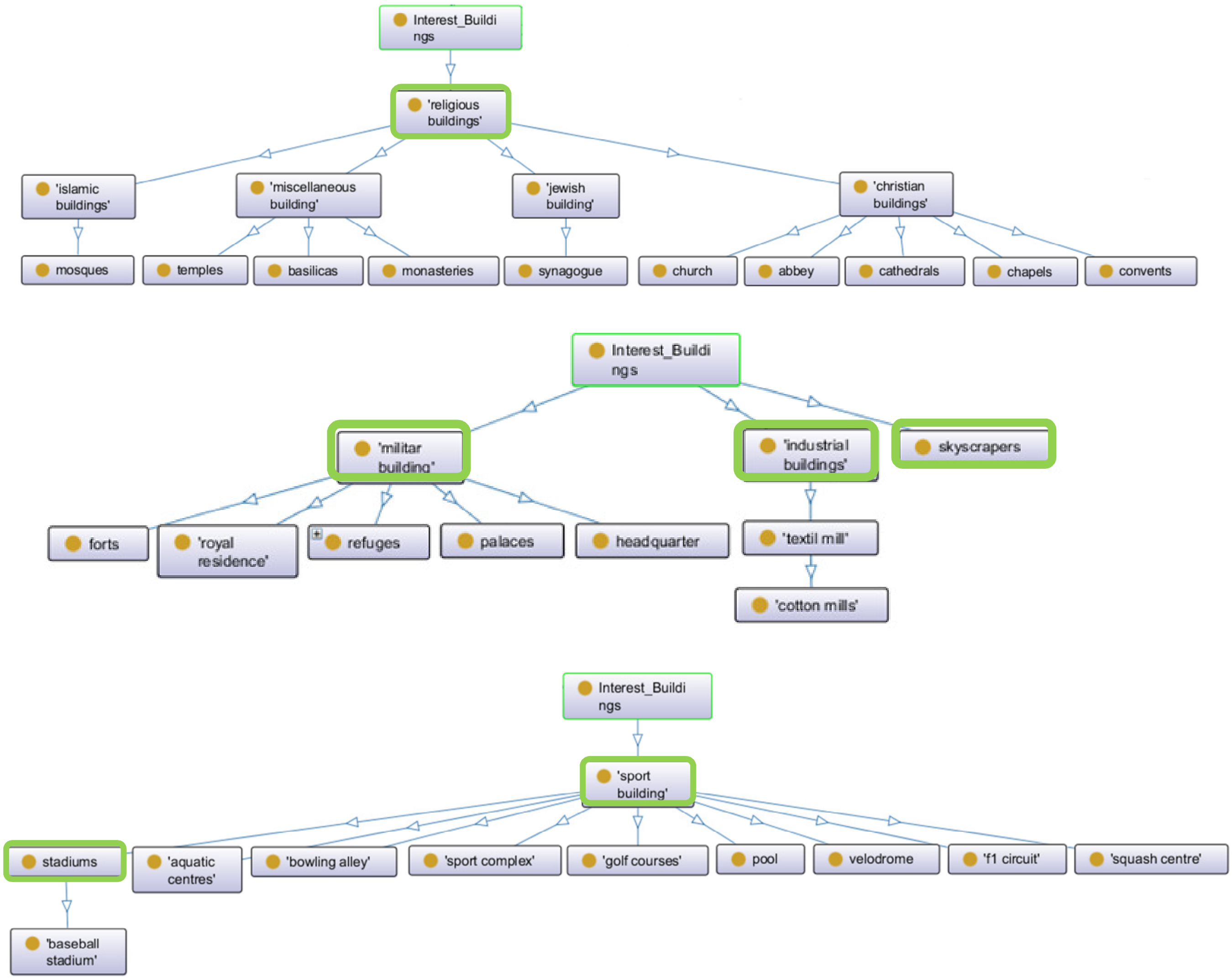


Figure 27. Block Interest buildings Ontology graph

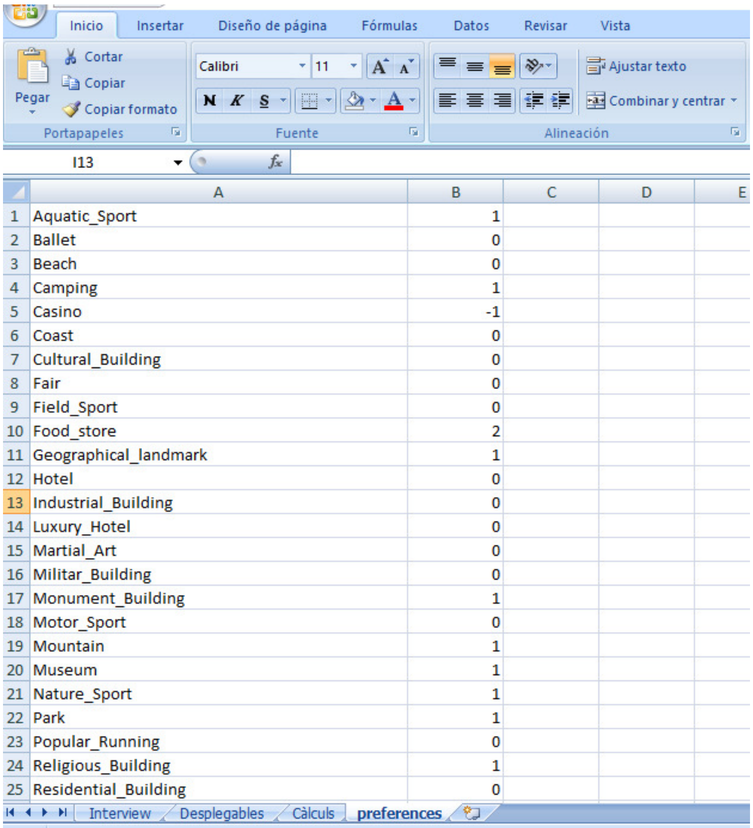
6.3.8 Implementation of the interview

I have developed the User profile interview with a Excel file. In the first sheet the DM answers about his preferences (see Figure 28).

GEOGRAPHICAL FEATURES			
Water landmarks (rivers, lakes,...)	*I don't care*		
Coast	*I don't care*		
Beaches	*I don't care*		
Geographical landmarks	*I like it*		
Mountains	*I like it*		
HOSTING			
Campings	*I like it*		
Hotels	*I don't care*		
Luxury hotels	*I don't care*		
Residential buildings	*I don't care*		
FACILITIES AND LEISURE			
Food stores	*Essential*		
Shopping	*I don't care*		
Shops	*I don't care*	Favourite shops	
Fairs	*I don't care*		
Casinos	*I don't like*		
Parks and gardens	*I like it*		
SPORTS			
Aquatic sports	*I like it*	Favourite aquatic sport	*Diving*
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*Indifferent*		
Field sports	*Indifferent*	Favourite field sport	
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*I like it*	Favourite nature sport	*Cycling*
CULTURE			
Museums	*I like it*	Favourite museums	
Cultural buildings	*I don't care*		
Monument buildings	*I like it*		
INTERESTING BUILDINGS			
Religious buildings	*I like it*		
Skyscrapers	*I like it*		
Industrial buildings	*I don't care*		
Militar buildings	*I don't care*		
Sport facilities	*I don't care*		
Stadiums	*I don't care*		

Figure 28. User profile interview.

In sheet *preferences* we obtain a list of class name and its preference value (see Figure 29).



	A	B
1	Aquatic_Sport	1
2	Ballet	0
3	Beach	0
4	Camping	1
5	Casino	-1
6	Coast	0
7	Cultural_Building	0
8	Fair	0
9	Field_Sport	0
10	Food_store	2
11	Geographical_landmark	1
12	Hotel	0
13	Industrial_Building	0
14	Luxury_Hotel	0
15	Martial_Art	0
16	Militar_Building	0
17	Monument_Building	1
18	Motor_Sport	0
19	Mountain	1
20	Museum	1
21	Nature_Sport	1
22	Park	1
23	Popular_Running	0
24	Religious_Building	1
25	Residential_Building	0

Figure 29. Preferences list.

The system will read this sheet to obtain the user profile.

6.4 Criteria

If we analyze the user profile interview, we obtain that the maximum number of criteria evaluated would be 38, distributed as follows:

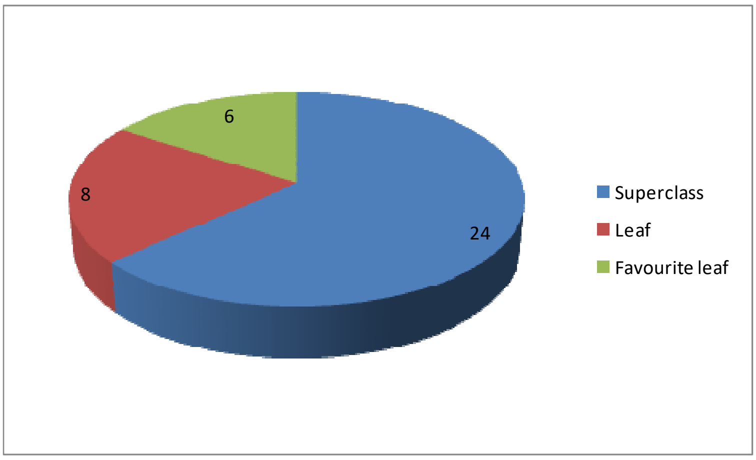


Figure 30. Criteria distribution.

Where Superclasses are classes with descendants, Leaves are classes without descendants and Favourite leaves are leaves that can be chosen as favourite descendant from a superclass.

From these 38 possible criteria, the system will only evaluate those that had an answer different than "*Indifferent*" by the DM.

When designing the questionnaire, I have tried to minimize the number of leaves in the interview in order to increase the effectiveness of the tests. But including some leaves is needed to make sense of the interview.

I have compared the ontology definition of the superclasses with the real data of the alternatives. Observing difference between the number of descendants in the ontology and the maximum punctuation in all alternatives we find different situations, such as the three cases shown in Table 10:

Criterion	**Ontology #Desc.**	**Alternatives Max.Value**	
Museum	28	10	In the ontology, there are defined 28 different kind of museums, but there is no city that has all of them. The maximum number of different museums a city has is 10.
Residential_Building	4	4	In the ontology, Residential_Building class has 4 descendant classes and there are some cities that have the four tags.
Hotel	1	2	Hotel has only a descendant: Luxury_Hotel. There are some cities that have both tags: Hotel and Luxury_Hotel. For this reason, these cities have a punctuation of two.

Table 10. Ontology definition and alternatives tags comparison.

The comparison between the number of descendants and the maximum number of values in the alternatives for all criteria is shown in Figure 31. We can observe significant differences in some classes. In general, the number of descendants is larger than the number of values in a single alternative, as it is reasonable.

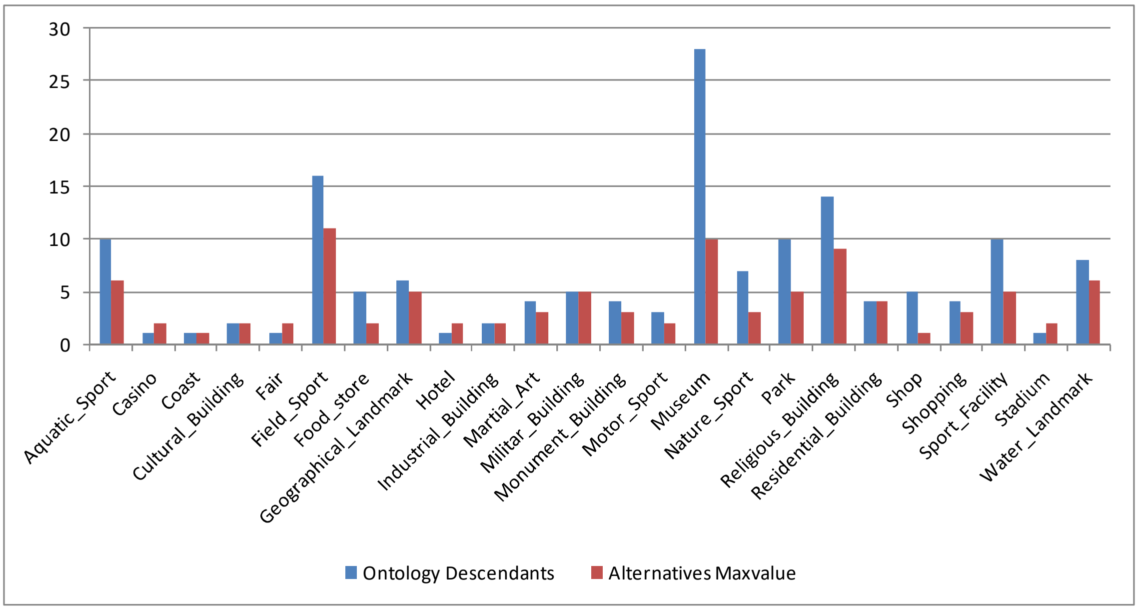


Figure 31. Difference between Ontology descendants and Alternatives maximum values for each criterion

6.5 Thresholds

As explained in section 5.6.1, Indifference and Preference thresholds will depend on two factors:

- The total number of descendants a class has.
- The distance from a class to its descendants.

Algorithm to calculate thresholds is as follows:

if (criterion is a leaf) then
    indifferenceThreshold = 0;
    preferenceThreshold = 0;
else
    *indifferenceThreshold =round(x*distanceToLeaves + y*numOfChildren);*
    preferenceThreshold = round(z distanceToLeaves + k*numOfChildren);*
    if (criterion is Essential) then
        vetoThreshold = preferenceThreshold + w;
    end if
end if

In order to determine *x, y, z, k* and *w* values I elaborated a table with all possible superclass criteria, their number of descendants and the distance to the last leaf in the ontology topology, the maximum performance value obtained for all alternatives and the threshold values obtained with different values of *x, y,z,k,w*.
I ran a high number of tests; only those with more relevant results are presented in this document.

Test 1: In first test I assigned the same values to *x* and *y* and the same values to *z* and *k*:

$x = 0.3$

$y = 0.3$

$z = 0.6$

$k = 0.6$

$w = 2$

In Figure 32 results of this test are shown. All invalid thresholds generated (i.e. the threshold value is higher than the maximum performance value in all alternatives) are marked in red. We can observe all veto thresholds are invalid, this indicates that *w* value is too high.

				Test 1		
Criteria name	Distance to leafs	Descend ontology	Max alternatives	Indifference threshold	Preference threshold	Veto threshold
Aquatic_Sport	2	10	6	4	7	9
Casino	1	1	2	1	1	3
Coast	1	1	1	1	1	3
Cultural_Building	1	2	2	1	2	4
Fair	1	1	2	1	1	3
Field_Sport	2	16	11	5	11	13
Food_store	2	5	2	2	4	6
Geographical_Landmark	1	6	5	2	4	6
Hotel	1	1	2	1	1	3
Industrial_Building	2	2	2	1	2	4
Martial_Art	1	4	3	2	3	5
Militar_Building	1	5	5	2	4	6
Monument_Building	2	4	3	2	4	6
Motor_Sport	2	3	2	2	3	5
Museum	2	28	10	9	18	20
Nature_Sport	2	7	3	3	5	7
Park	2	10	5	4	7	9
Religious_Building	2	14	9	5	10	12
Residential_Building	1	4	4	2	3	5
Shop	2	5	1	2	4	6
Shopping	1	4	3	2	3	5
Sport_Facility	2	10	5	4	7	9
Stadium	1	1	2	1	1	3
Water_Landmark	2	8	6	3	6	8

Figure 32. Test 1 to determine x, y, z, k, w values

Test 2: In this test I reduced all the values (*x, y, z, k, w)* in order to do not obtain invalid thresholds:

$x = 0.1$

$y = 0.1$

$z = 0.2$

$k = 0.2$

$w = 1$

In Figure 33 results of this test are shown. We can observe there are still two invalid thresholds (marked in dark red). Moreover, we can observe another problem: for some criteria, indifference and preference threshold are the same value (this situation are marked in light red). With a visual examination of thresholds values obtained we can observe there are some indifference thresholds that could be a slightly high. For instance, criterion *Food store* can have a maximum performance value of 2, but it has an indifference threshold of 1.

				Test 2		
Criteria name	Distance to leafs	Descend ontology	Max alternatives	Indifference threshold	Preference threshold	Veto threshold
Aquatic_Sport	2	10	6	1	2	3
Casino	1	1	2	0	0	1
Coast	1	1	1	0	0	1
Cultural_Building	1	2	2	0	1	2
Fair	1	1	2	0	0	1
Field_Sport	2	16	11	2	4	5
Food_store	2	5	2	1	1	2
Geographical_Landmark	1	6	5	1	1	2
Hotel	1	1	2	0	0	1
Industrial_Building	2	2	2	0	1	2
Martial_Art	1	4	3	1	1	2
Militar_Building	1	5	5	1	1	2
Monument_Building	2	4	3	1	1	2
Motor_Sport	2	3	2	1	1	2
Museum	2	28	10	3	6	7
Nature_Sport	2	7	3	1	2	3
Park	2	10	5	1	2	3
Religious_Building	2	14	9	2	3	4
Residential_Building	1	4	4	1	1	2
Shop	2	5	1	1	1	2
Shopping	1	4	3	1	1	2
Sport_Facility	2	10	5	1	2	3
Stadium	1	1	2	0	0	1
Water_Landmark	2	8	6	1	2	3

Figure 33. Test 2 to determine x, y, z, k, w values

Test 3: As shown in test 2 results, threshold values were still slightly high. In Test 3, I reduced *x, y, z, k* values. In this test, *x* and *z* will be proportional to *y* and *k,* respectively in the same proportion as the median of the distance and the median number of alternatives. We have:

median(distance) = 2

median(maxPerformance) = 3

median(distance) / median(distance) = 1.5

Consequently $x = 1.5*y$ and z = 1.5*k

***x* = 0.075**

***y* = 0.05**

***z* = 0.225**

***k* = 0.15**

***w* = 1**

In Figure 34 thresholds obtained in this test are shown. We can observe there is only one invalid threshold for criterion *Shop*. This criterion has an exclusive character as, although it has five descendants in the ontology definition, maximum performance a city has is one. In fact, this criterion is acting as a leaf and essential leafs do not have veto threshold. In this test moreover, we do not find the problem of having equal indifference and preference thresholds.

Finally these values are chosen to run the proposed solution tests as are the ones that generate best thresholds.

				Test 3		
Criteria name	**Distance to leafs**	**Descend ontology**	**Max alternatives**	**Indifference threshold**	**Preference threshold**	**Veto threshold**
Aquatic_Sport	2	10	6	1	2	3
Casino	1	1	2	0	0	1
Coast	1	1	1	0	0	1
Cultural_Building	1	2	2	0	1	2
Fair	1	1	2	0	0	1
Field_Sport	2	16	11	1	3	4
Food_store	2	5	2	0	1	2
Geographical_Landmark	1	6	5	0	1	2
Hotel	1	1	2	0	0	1
Industrial_Building	2	2	2	0	1	2
Martial_Art	1	4	3	0	1	2
Militar_Building	1	5	5	0	1	2
Monument_Building	2	4	3	0	1	2
Motor_Sport	2	3	2	0	1	2
Museum	2	28	10	2	5	6
Nature_Sport	2	7	3	1	2	3
Park	2	10	5	1	2	3
Religious_Building	2	14	9	1	3	4
Residential_Building	1	4	4	0	1	2
Shop	2	5	1	0	1	2
Shopping	1	4	3	0	1	2
Sport_Facility	2	10	5	1	2	3
Stadium	1	1	2	0	0	1
Water_Landmark	2	8	6	1	2	3

Figure 34. Test 3 to determine x, y, z, k, w values

7 Results

In order to analyse and compare the results obtained, I have created 4 different profiles: the *Fashion Victim*, the *Beach Lover*, the *Culture Freak* and mine: the *Big Family*. In the next sections of this chapter the results obtained with each profile are explained in detail.

7.1 Fashion Victim profile

7.1.1 Description

In this example, the user is a group of women that are planning to do a shopping travel. For this reason, they have marked as essential the classes *Shops* and *Shopping*. They also like musical and dance shows, for this reason they have marked as *like it* the criteria *Ballet* and *Cultural Buildings*.

GEOGRAPHICAL FEATURES			
Water landmarks (rivers, lakes,...)	*Indifferent*		
Coast	*I like it*		
Beaches	*I like it*		
Geographical landmarks	*Indifferent*		
Mountains	*Indifferent*		
HOSTING			
Campings	*Indifferent*		
Hotels	*I like it*		
Luxury hotels	*I like it*		
Residential buildings	*Indifferent*		
FACILITIES AND LEISURE			
Food stores	*Indifferent*		
Shopping	*Essential*		
Shops	*Essential*	Favourite shops	*Fashion shops*
Fairs	*Indifferent*		
Casinos	*Indifferent*		
Parks and gardens	*I like it*		
SPORTS			
Aquatic sports	*Indifferent*	Favourite aquatic sport	
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*I like it*		
Field sports	*Indifferent*	Favourite field sport	
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*Indifferent*	Favourite nature sport	
CULTURE			
Museums	*Indifferent*	Favourite museums	
Cultural buildings	*I like it*		
Monument buildings	*Indifferent*		
INTERESTING BUILDINGS			
Religious buildings	*Indifferent*		
Skyscrapers	*I like it*		
Industrial buildings	*Indifferent*		
Militar buildings	*I don't like it*		
Sport facilities	*Indifferent*		
Stadiums	*Indifferent*		

Figure 35. Fashion Victim Profile Preferences

7.1.2 Criteria

The number of criteria that have to be evaluated is reduced to 12, only classes that have been evaluated by the DM in the preferences interview: *Ballet*, *Beach*, *Coast*, *Cultural Building*, *Hotel*, *Luxury Hotel*, *Military Building*, *Park*, *Shop*, *Shopping*, *Skyscraper* and *Fashion Shop*. All of them have to be maximized except *Military Building* in which the DM have expressed dislike.

Name	maximize / minimize
Ballet	maximize
Beach	maximize
Coast	maximize
Cultural building	maximize
Hotel	maximize
Luxury Hotel	maximize
Military Building	minimize
Park	maximize
Shop	maximize
Shopping	maximize
Skyscraper	maximize
Fashion Shop	maximize

Table 11. Fashion Victim Profile Criteria Set.

7.1.3 Alternatives

First of all, as the classes *Shopping* and *Shop* are essential, cities that have zero tags in the branch shopping or zero in the branch shop are removed from the set of alternatives. As show in Figure 36 candidate cities have to accomplish at least with one of these features: *Shopping*, *Shopping areas*, *Shopping street*, *Shopping centres* or *Malls* and at least one of these: *Shop*, *Souvenir shops*, *Antiquarian*, *Clothes shop*, *Outlets* or *Fashion shops*.

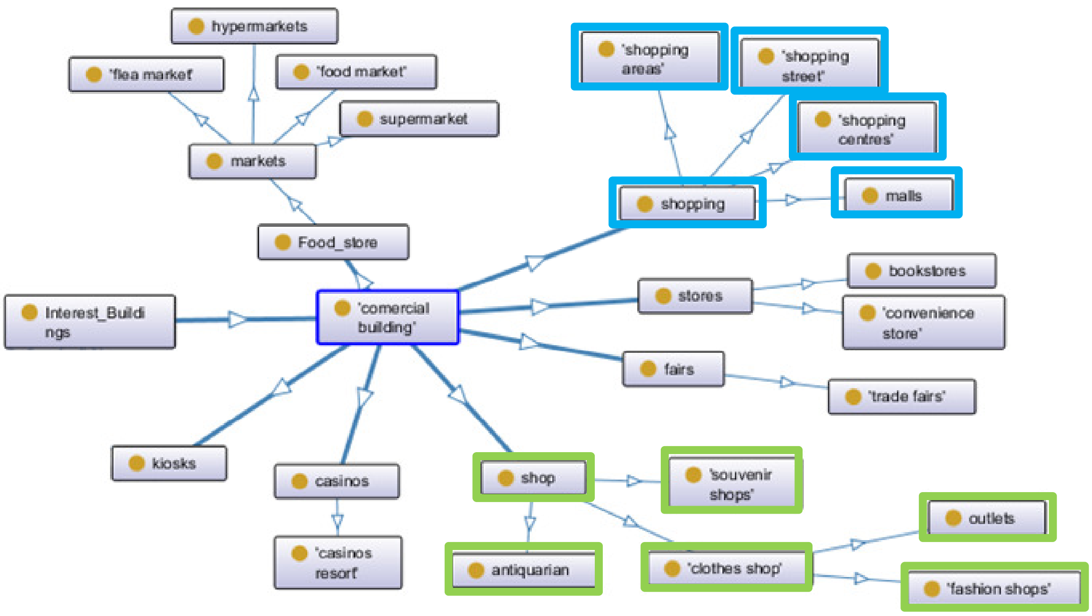


Figure 36. Subclasses of shopping and shop.

After first step, the alternatives' set is reduced to 33 cities, shown in Figure 37.

Figure 37. Set of alternatives for Fashion Victim profile.

7.1.4 Performance table

The process to obtain the performance table is very simple: just counting how many tags of the alternative are in the list of subclasses depending on each criterion (see Table 12).

	Ballet	Beach	Coast	Cultural Building	Hotel	Luxury Hotel	Military Building	Park	Shop	Shopping	Skyscraper	Fashion Shop
Amsterdam	1	0	0	2	1	0	3	2	1	1	0	0
Bath Somerset	0	1	1	2	2	1	2	2	1	2	0	0
Benidorm	0	1	1	0	1	0	0	2	1	1	1	0
Birminghan	1	0	0	2	1	0	1	2	1	1	0	0
Buenos Aires	0	0	0	2	1	0	2	3	1	2	1	0
Cairo	0	1	1	2	1	0	1	1	1	1	0	0
Cape Town	0	1	1	1	0	0	0	1	1	1	0	0
Cardiff	1	1	1	2	1	0	2	2	1	3	1	0
Chester	0	0	0	2	1	0	1	2	1	1	0	0
Chicago	1	1	1	2	1	0	3	4	1	2	1	0
Dresden	1	0	0	2	1	0	3	2	1	1	0	0
Dublin	0	0	0	2	1	0	2	1	1	2	0	0
Edinburgh	0	0	0	2	0	0	3	2	1	2	1	0
Florence	0	0	0	1	0	0	3	0	1	1	1	0
Genoa	0	0	0	2	0	0	1	1	1	2	0	0
Havana	1	1	1	2	1	1	3	1	1	1	0	0
Honolulu	0	1	1	2	1	0	2	2	1	2	0	0
Houston	1	0	0	2	1	0	2	3	1	1	1	0
Istanbul	1	1	1	2	1	0	2	1	1	1	0	0
London	1	0	0	2	0	0	3	2	1	2	1	0
Milan	1	0	0	2	0	0	1	1	1	3	1	0
Moscow	0	0	0	1	1	0	3	4	1	2	1	0
Munich	1	0	0	2	1	0	3	1	1	1	1	0
Nottingham	0	0	0	1	1	0	0	1	1	2	0	0
Orlando	0	1	1	1	1	0	1	2	1	1	1	0
Oxford	0	0	0	1	1	0	0	1	1	1	0	0
Paris	1	0	0	2	1	0	3	3	1	3	1	0
Reading Pennsylvania	0	0	0	0	0	0	1	1	1	1	0	0
San Francisco	1	1	1	2	1	0	4	3	1	1	1	0
Sao Paulo	1	0	0	2	1	0	3	1	1	2	1	0
Seattle	1	1	1	2	1	0	1	2	1	1	1	0
Seoul	0	0	0	0	1	0	3	1	1	2	1	0
Zurich	1	0	0	2	1	0	2	2	1	2	0	1

Table 12. Fashion Victim profile performance table.

7.1.5 Weights

In this test we have three kind of weights: liked or disliked leaf (1x), superclass (2x) and essential superclass (3x). First, the process calculates x:

$$x = \frac{1}{numLeafs + (2 * numSuperclass) + (3 * numEssentialSuperclass)} = 0.0476$$

where:

- *numLeaves = 5 (Ballet, Beach, Luxury Hotel, Skyscraper* and *Fashion shop*).
- *numSuperclass = 5 (Coast, Cultural building, Hotel, Military building* and *Park).*
- *numEssentialSuperclass = 2 (Shop* and *Shopping).*

Criteria	Weight
Ballet	0.047619047619047616
Beach	0.047619047619047616
Coast	0.09523809523809523
Cultural Building	0.09523809523809523
Hotel	0.09523809523809523
Luxury Hotel	0.047619047619047616
Militar Building	0.09523809523809523
Park	0.09523809523809523
Shop	0.14285714285714285
Shopping	0.14285714285714285
Skyscraper	0.047619047619047616
Fashion Shop	0.047619047619047616

Table 13. Fashion Victim profile Weights

7.1.6 Thresholds

In Table 14 we can see a comparison between thresholds generated for *Fashion Victim* profile with *Method 1* and *Method 2*. Let us remember that *Method 1* generates thresholds taking into account the number of descendants defined in the ontology while *Method 2* generates thresholds taking into account the max value for the criteria found in the alternatives set of the test (alternatives removed from the set are not taken into account).

Following, we have the table legend:

- *Criteria*: criterion name.
- *Is a leaf*: the value is No if the criterion has descendants and Yes otherwise.
- *Distance to leaf*: maximum number of steps to arrive to all descendants of the class.
- *Ontology #Descend.*: number of descendants of the class considering the ontology definition.
- *Maxvalue all altern.*: max number of tags for the criterion found in the whole set of alternatives.
- *Maxvalue in test*: max number of tags for the criterion found in the alternatives that are considered for this test (alternatives with performance > 0 for the essential tags).
- *Indifference Threshold*: indifference thresholds calculated with method 1 and 2. Marked in red when are not identical.

- *Preference Threshold*: preference thresholds calculated with method 1 and 2. Marked in red when are not identical.
- *Veto Threshold*: indifference thresholds calculated with method 1 and 2. If it is empty in the table, it means that there is no veto threshold for this criterion.

						Method 1			Method 2		
Criteria	Is a leaf	Distance to leaf	Ontology #Descend.	Maxvalue all altern.	Maxvalue in test	Indifference Threshold	Preference Threshold	Veto Threshold	Indifference Threshold	Preference Threshold	Veto Threshold
Coast	No	1	1	1	1	0	0		0	0	
Cultural Building	No	1	2	2	2	0	1		0	1	
Hotel	No	1	1	2	2	0	0		0	1	
Militar Building	No	1	5	5	4	0	1		0	1	
Park	No	2	10	5	4	1	2		0	1	
Shop	No	2	5	1	1	0	1	2	0	1	2
Shopping	No	1	4	3	3	0	1	2	0	1	2
Ballet	Yes				1	0	0		0	0	
Beach	Yes				1	0	0		0	0	
Fashion Shop	Yes				1	0	0		0	0	
Luxury Hotel	Yes				1	0	0		0	0	
Skyscraper	Yes				1	0	0		0	0	

Table 14. Fashion Victim Profile Thresholds Analysis

7.1.7 Touristic destinations ranking

In Figure 39 and Figure 38 we observe the touristic destination ranking obtained with Electre Distillation using thresholds calculation methods 1 and 2, respectively.

Ranking obtained is quite similar using both methods, the reason of this similarity is that thresholds obtained with both methods are quite similar. Best city in both cases is Cardiff and worst is Florence. In Table 15 I show graphically what would be best and worst cities in the ranking. I have marked in dark green best punctuations for each criterion and in light green the second ones. I have marked in dark and light red worst and second worst punctuations for criterion that have to be minimized.

As we can quickly observe, Cardiff has the best punctuations in all criteria except in *Luxury Hotel*, *Park* and *Fashion Shop*. Next best cities are Chicago, Seattle and Bath (in Somerset). Chicago is better than Cardiff in *Park*, but it is worse in *Shopping* and *Militar Building. Shopping* has a higher weight because it is an essential criterion. This fact makes Cardiff better than Chicago. Seattle and Bath have also worse punctuations in Shopping than Cardiff.

At first sight I had never thought in Cardiff as a Shopping city. I had probably thought in New York, Paris or Milan. But New York is not in the alternatives set because it has not the tag *Shop*, that was an essential one. Paris and Milan are in the ranking but not as well positioned as Cardiff because they have no coast.

In wikipedia information about Cardiff we find following sentences that confirm the results:

- *The city is the country's chief commercial centre, the base for most national cultural...*
- *Cardiff is a significant tourist centre and the most popular visitor destination in Wales...*
- *In 2011, Cardiff was ranked sixth in the world in National Geographic's alternative tourist destinations...*

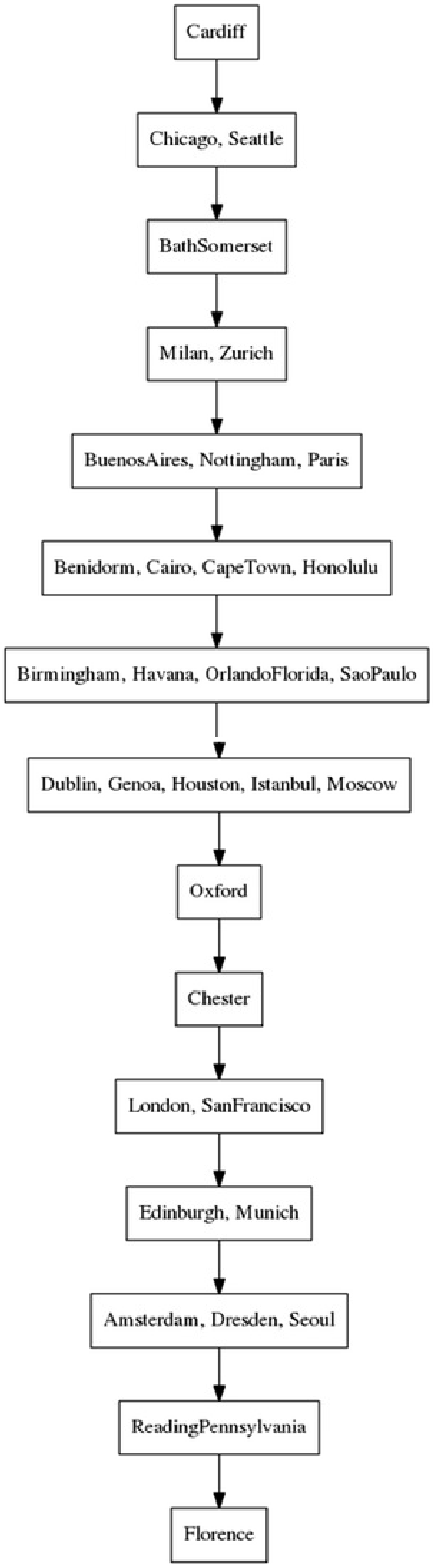


Figure 39. Fashion Victim ranking with Method1

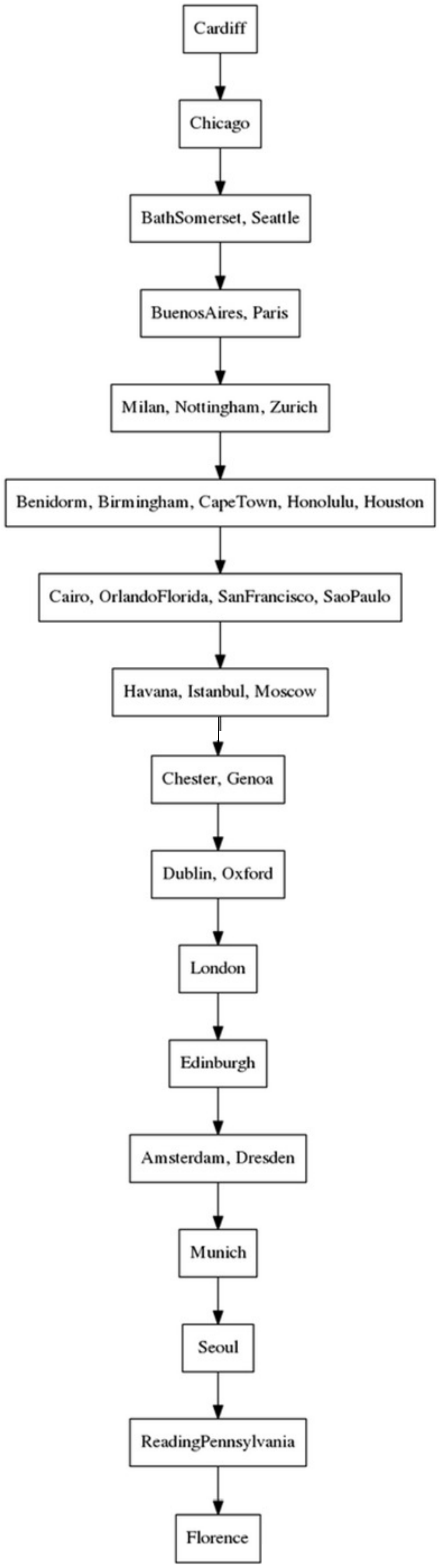


Figure 38. Fashion Victim ranking with Method2

	Ballet	Beach	Coast	Cultural Building	Hotel	Luxury Hotel	Militar Building	Park	Shop	Shopping	Skyscraper	Fashion Shop
Amsterdam	1	0	0	2	1	0	3	2	1	1	0	0
Bath Somerset	0	1	1	2	2	1	2	2	1	2	0	0
Benidorm	0	1	1	0	1	0	0	2	1	1	1	0
Birminghan	1	0	0	2	1	0	1	2	1	1	0	0
Buenos Aires	0	0	0	2	1	0	2	3	1	2	1	0
Cairo	0	1	1	2	1	0	1	1	1	1	0	0
Cape Town	0	1	1	1	0	0	0	1	1	1	0	0
Cardiff	1	1	1	2	1	0	2	2	1	3	1	0
Chester	0	0	0	2	1	0	1	2	1	1	0	0
Chicago	1	1	1	2	1	0	3	4	1	2	1	0
Dresden	1	0	0	2	1	0	3	2	1	1	0	0
Dublin	0	0	0	2	1	0	2	1	1	2	0	0
Edinburgh	0	0	0	2	0	0	3	2	1	2	1	0
Florence	0	0	0	1	0	0	3	0	1	1	1	0
Genoa	0	0	0	2	0	0	1	1	1	2	0	0
Havana	1	1	1	2	1	1	3	1	1	1	0	0
Honolulu	0	1	1	2	1	0	2	2	1	2	0	0
Houston	1	0	0	2	1	0	2	3	1	1	1	0
Istanbul	1	1	1	2	1	0	2	1	1	1	0	0
London	1	0	0	2	0	0	3	2	1	2	1	0
Milan	1	0	0	2	0	0	1	1	1	3	1	0
Moscow	0	0	0	1	1	0	3	4	1	2	1	0
Munich	1	0	0	2	1	0	3	1	1	1	1	0
Nottingham	0	0	0	1	1	0	0	1	1	2	0	0
Orlando	0	1	1	1	1	0	1	2	1	1	1	0
Oxford	0	0	0	1	1	0	0	1	1	1	0	0
Paris	1	0	0	2	1	0	3	3	1	3	1	0
Reading Pennsylvania	0	0	0	0	0	0	1	1	1	1	0	0
San Francisco	1	1	1	2	1	0	4	3	1	1	1	0
Sao Paulo	1	0	0	2	1	0	3	1	1	2	1	0
Seattle	1	1	1	2	1	0	1	2	1	1	1	0
Seoul	0	0	0	0	1	0	3	1	1	2	1	0
Zurich	1	0	0	2	1	0	2	2	1	2	0	1

Table 15. Fashion Victim Performance table with colour gradation.

7.2 Beach Lover profile

7.2.1 Description

These are a couple that love windsurf, so they consider essential *Coast, Beaches* and *Aquatic Sports*. As they want to stay for some days in a camping they have marked as liked *Camping* and *Food stores*. They also like *Water landmarks, Parks, Nature sports* and *Sport facilities.* And they do not like *Industrial buildings.*

GEOGRAPHICAL FEATURES			
Water landmarks (rivers, lakes,...)	*I like it*		
Coast	*Essential*		
Beaches	*Essential*		
Geographical landmarks	*Indifferent*		
Mountains	*Indifferent*		
HOSTING			
Campings	*I like it*		
Hotels	*Indifferent*		
Luxury hotels	*Indifferent*		
Residential buildings	*Indifferent*		
FACILITIES AND LEISURE			
Food stores	*I like it*		
Shopping	*Indifferent*		
Shops	*Indifferent*	Favourite shops	
Fairs	*Indifferent*		
Casinos	*Indifferent*		
Parks and gardens	*I like it*		
SPORTS			
Aquatic sports	*Essential*	Favourite aquatic sport	*Windsurfing*
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*Indifferent*		
Field sports	*Indifferent*	Favourite field sport	
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*I like it*	Favourite nature sport	
CULTURE			
Museums	*Indifferent*	Favourite museums	
Cultural buildings	*Indifferent*		
Monument buildings	*Indifferent*		
INTERESTING BUILDINGS			
Religious buildings	*Indifferent*		
Skyscrapers	*Indifferent*		
Industrial buildings	*I don't like it*		
Militar buildings	*Indifferent*		
Sport facilities	*I like it*		
Stadiums	*Indifferent*		

Figure 40. Beach Lover Profile Interview

7.2.2 Criteria

The number of criteria that have to be evaluated is reduced to 11, only classes that have been evaluated by the DM in the preferences interview: Aquatic *Sport, Beach, Camping, Coast, Food Store, Industrial Building, Nature Sport, Park, Sport Facility, Water Landmark* and *Windsurfing*. All of them have to be maximized except *Industrial Building* in which the DM has expressed dislike.

Name	maximize / minimize
Aquatic Sport	maximize
Beach	maximize
Camping	maximize
Coast	maximize
Food store	maximize
Industrial Building	minimize
Nature Sport	maximize
Park	maximize
Sport Facility	maximize
Water Landmark	maximize
Windsurfing	maximize

Table 16. Beach Lover Profile Criteria Set.

7.2.3 Alternatives

Cities that have zero tags in the groups *Aquatic sport* or *Coast* or in the leaf *Beach* are removed from the set of alternatives. As show in Figure 41 candidate cities have to accomplish at least with one of these features: *Aquatic sport, Surfing, Water Surfing, Windsurfing, Swimming, Sailing, Kayaking, Rafting, Water Polo, Under Water* or *Diving* and at least one of these: *Coast* or *Beach* and mandatory have to accomplish the leaf *Beach.*

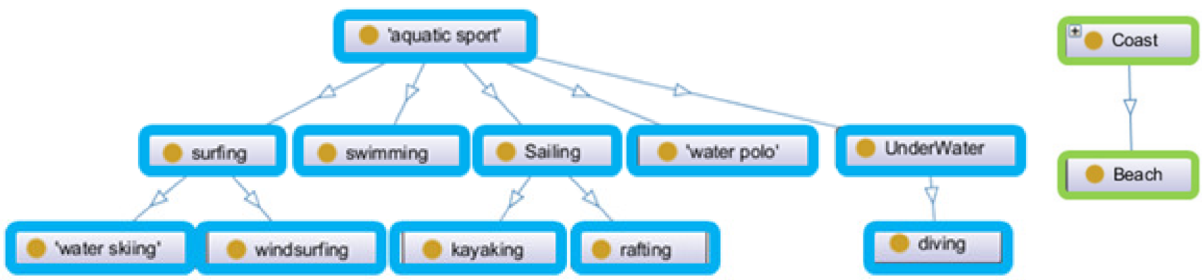


Figure 41. Essential classes of Beach Lover profile.

After first step, our alternatives' set is reduced to 33 cities, shown in Figure 42.

Figure 42. Set of alternatives for Beach Lover profile.

7.2.4 Performance table

In Table 17 we can see Performance Table generated for Beach Lover profile.

	Aquatic Sport	Beach	Camping	Coast	Food store	Industrial Building	Nature Sport	Park	Sport Facility	Water Landmark	Windsurfing
Aberdeen	1	1	0	1	1	0	2	1	3	2	0
Abu Dhabi	1	1	0	1	1	0	0	1	0	1	0
Barcelona	1	1	0	1	1	1	0	5	2	1	0
Bath Somerset	1	1	0	1	1	0	1	2	2	4	0
Brighton	1	1	0	1	0	0	0	1	2	2	0
Bruges	1	1	0	1	1	0	1	1	1	3	0
Cancun	1	1	0	1	0	0	0	1	0	4	0
Cape Town	2	1	0	1	0	0	2	1	1	1	0
Cardiff	3	1	0	1	1	0	3	2	2	4	0
Chicago	1	1	0	1	1	0	1	4	1	5	0
Copenhagen	2	1	0	1	1	1	1	2	1	4	0
Dalian	1	1	0	1	0	0	1	1	1	3	0
Florianopolis	2	1	0	1	1	0	3	0	1	3	0
Fortaleza	2	1	0	1	1	0	0	1	1	4	1
Glasgow	1	1	0	1	1	0	1	1	2	4	0
Hamburg	1	1	0	1	1	0	1	2	1	5	0
Helsinki	1	1	0	1	1	0	0	1	2	2	0
Hong Kong	2	1	0	1	1	0	0	3	1	2	0
Istanbul	2	1	0	1	1	0	2	1	1	3	0
Kuala Lumpur	1	1	0	1	1	0	1	3	4	4	0
Malmo	1	1	0	1	1	0	0	0	1	1	0
Marseille	3	1	0	1	1	0	1	1	4	2	1
Qingdao	1	1	0	1	0	0	0	3	2	2	0
Reykjavik	1	1	0	1	1	0	0	1	1	2	0
Rio de Janeiro	2	1	0	1	0	0	1	1	1	4	0
Salvador Bahia	3	1	0	1	1	0	2	1	2	3	0
San Diego	2	1	0	1	1	0	1	3	1	2	0
San Francisco	2	1	0	1	1	0	1	3	2	4	1
Seattle	3	1	0	1	1	0	3	2	1	4	0
Sydney	2	1	0	1	1	0	0	2	1	4	0
Valencia	1	1	0	1	1	0	0	1	0	3	0
Varadero	1	1	0	1	0	0	0	2	0	1	0
Venice	1	1	0	1	0	0	0	1	0	6	0

Table 17. Beach Lover profile performance table.

7.2.5 Weights

In this test we have the four kind of weights: essential leaf (0x), liked or disliked leaf (1x), superclass (2x) and essential superclass (3x). As we can check out in Table 17 all alternatives have *Beach* tag because it is essential and it is a leaf, in consequence it will not be evaluated.

First, the process calculates x:

$$x = \frac{1}{numLeafs + (2 * numSuperclass) + (3 * numEssentialSuperclass)} = 0.05$$

where:

- *numLeaves = 2 (Windsurfing, Camping).*
- *numSuperclass = 6 (Food store, Industrial Building, Nature sport, Park, Sport facility, Water Landmark).*
- *numEssentialSuperclass = 2 (Aquatic sport, Coast).*

Criteria	Weight
Aquatic Sport	0.15
Beach	0.0
Camping	0.05
Coast	0.15
Food store	0.1
Industrial Building	0.1
Nature Sport	0.1
Park	0.1
Sport Facility	0.1
Water Landmark	0.1
Windsurfing	0.05

Table 18. Beach Lover profile Weights

7.2.6 Thresholds

In Table 19 we can see the difference between thresholds calculated with method 1 (taking into account the otology definition) and with method 2 (taking into account max performance values of the alternatives set). Max values in alternatives are equal or smaller than the number of descendants in the ontology definition, consequently thresholds are also less tolerant to differences when calculated with max value in alternatives. This causes that with method 2 all indifference thresholds are 0.

						Method 1			Method 2		
Criteria	Is a leaf	Distance to leaf	Ontology #Descend.	Maxvalue all altern.	Maxvalue in test	Indifference Threshold	Preference Threshold	Veto Threshold	Indifference Threshold	Preference Threshold	Veto Threshold
Aquatic Sport	No	2	10	6	3	1	2	3	0	1	2
Coast	No	1	1	1	1	0	0	1	0	0	1
Food store	No	2	5	2	1	0	1		0	1	
Industrial Building	No	2	2	2	1	0	1		0	1	
Nature Sport	No	2	7	3	3	1	2		0	1	
Park	No	2	10	5	5	1	2		0	1	
Sport Facility	No	2	10	5	4	1	2		0	1	
Water Landmark	No	2	8	6	6	1	2		0	1	
Beach	Yes			1	1	0	0	1	0	0	1
Camping	Yes			1	0	0	0		0	0	
Windsurfing	Yes			1	1	0	0		0	0	

Table 19. Beach Lover profile thresholds

7.2.7 Touristic destinations ranking

In Figure 44 and Figure 43 we can observe the rankings obtained with Method 1 and Method 2. In this test results are slightly different due to the different thresholds applied.

Let us analyse in detail the origin of the difference in ranking between the two best positioned alternatives Cardiff and San Francisco under the method used.

Cardiff and San Francisco have the same performance in all criteria except *Aquatic sport, Nature sport, Park* and *Windsurf:*

	Cardiff	San Francisco
Aquatic sport	3	2
Nature sport	3	1
Park	2	3
Windsurfing	0	1

Table 20. Different preferences in alternatives Cardiff and San Francisco.

With Method 1 both cities are indifferent in *Aquatic sport* and *Park* because indifference threshold is 1. In *Nature sport* Cardiff is not strictly preferred to San Francisco because preference threshold is 2 and in *Windsurfing* San Francisco is strictly preferred to Cardiff. Consequently San Francisco wins.

With method 2 we have not indifference threshold, consequently Cardiff is strictly preferred to San Francisco in *Nature sport* and weakly preferred in *Aquatic sport.* By the other hand, San Francisco is weakly preferred in *Park* and strictly preferred in *Windsurfing.* It seams there is a tie, but *Aquatic sport* weight is higher, so Cardiff wins.

It is surprising that Cardiff is one of the best positioned cities in this test again. If we analyse in detail why Cardiff is a such interesting tourism destiny, we find that in DAMASK City Matrix, Cardiff has the maximum number of tags: 60. This number is more than twice the median of the number of tags, which is 29. This means that, when comparing cities, Cardiff has more chances to have a better punctuation. This difference between number of tags is probably due to the fact that DAMASK city matrix was obtained with automatic processes that obtained information from the Web. In the moment of obtaining data, probably Cardiff had a very good web page with all its features detailed.

At first sight it seems that English-speaking cities are the best positioned regarding their number of tags, but I have not checked this. This could be explained by the fact that automatic processes to obtain DAMASK City Matrix parsed English web pages.

This test shows that, when the quality of information about alternatives is not uniform, it is better to use Method 1, based in the Ontology definition. Method 1 could be considered a more objective method.

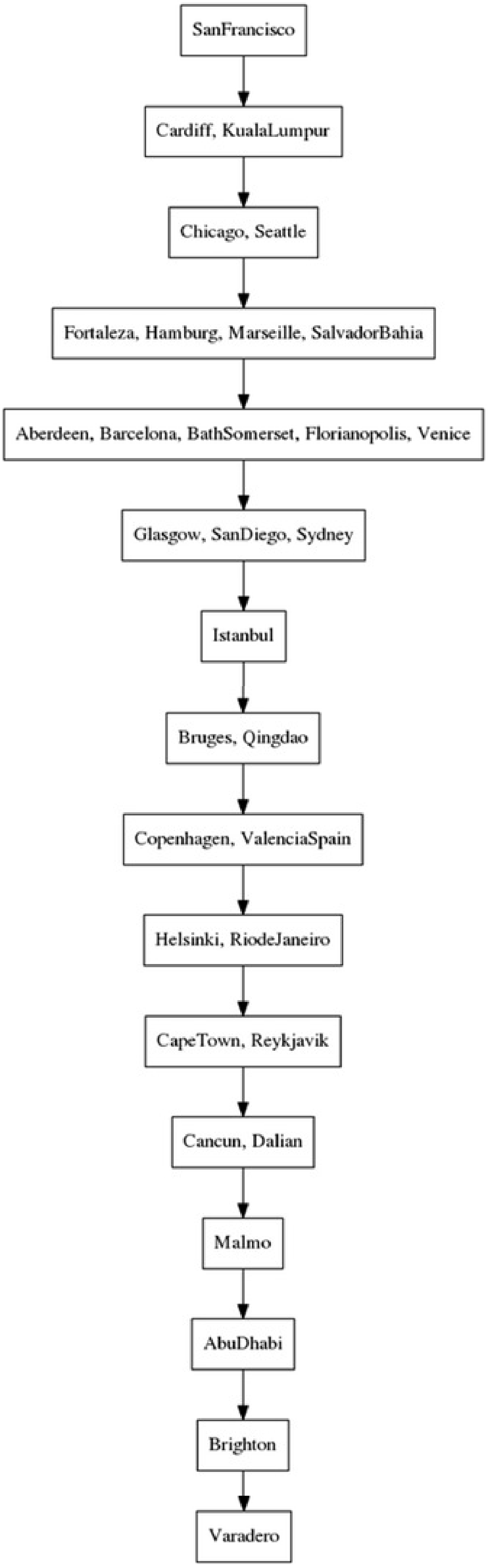


Figure 44. Beach Lover city ranking with Method 1

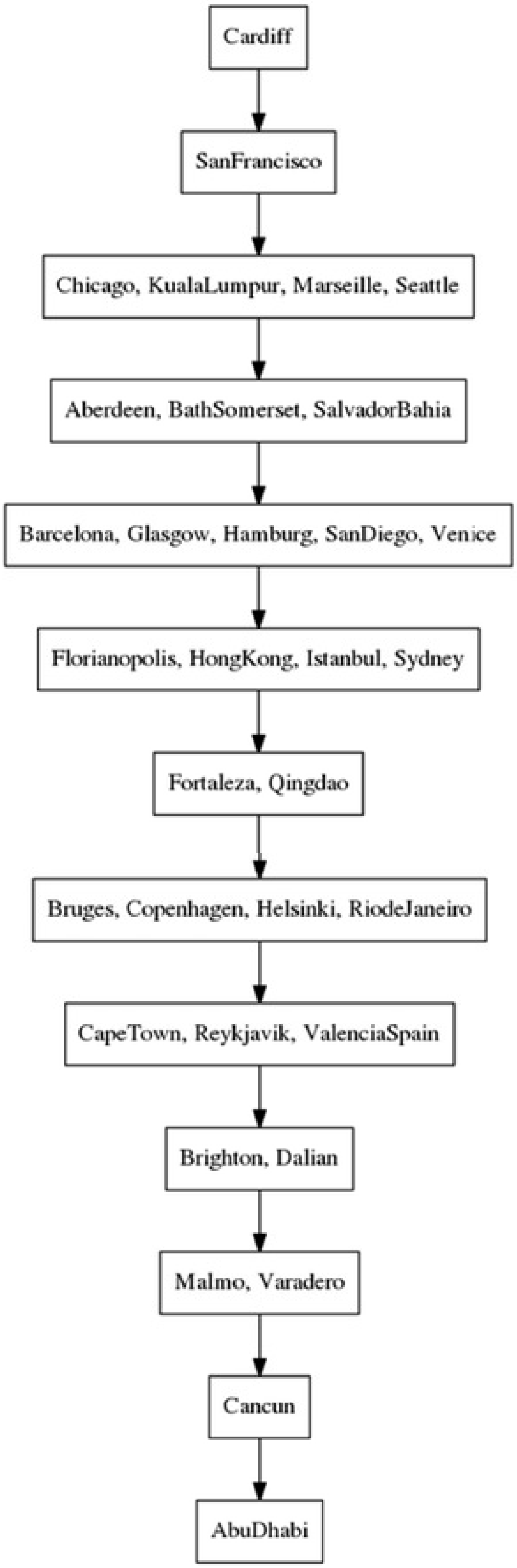


Figure 43. Beach Lover city ranking with Method 2

	Aquatic Sport	Beach	Camping	Coast	Food store	Industrial Building	Nature Sport	Park	Sport Facility	Water Landmark	Windsurfing
Aberdeen	1	1	0	1	1	0	2	1	3	2	0
Abu Dhabi	1	1	0	1	1	0	0	1	0	1	0
Barcelona	1	1	0	1	1	1	0	5	2	1	0
Bath Somerset	1	1	0	1	1	0	1	2	2	4	0
Brighton	1	1	0	1	0	0	0	1	2	2	0
Bruges	1	1	0	1	1	0	1	1	1	3	0
Cancun	1	1	0	1	0	0	0	1	0	4	0
Cape Town	2	1	0	1	0	0	2	1	1	1	0
Cardiff	3	1	0	1	1	0	3	2	2	4	0
Chicago	1	1	0	1	1	0	1	4	1	5	0
Copenhagen	2	1	0	1	1	1	1	2	1	4	0
Dalian	1	1	0	1	0	0	1	1	1	3	0
Florianopolis	2	1	0	1	1	0	3	0	1	3	0
Fortaleza	2	1	0	1	1	0	0	1	1	4	1
Glasgow	1	1	0	1	1	0	1	1	2	4	0
Hamburg	1	1	0	1	1	0	1	2	1	5	0
Helsinki	1	1	0	1	1	0	0	1	2	2	0
Hong Kong	2	1	0	1	1	0	0	3	1	2	0
Istanbul	2	1	0	1	1	0	2	1	1	3	0
Kuala Lumpur	1	1	0	1	1	0	1	3	4	4	0
Malmo	1	1	0	1	1	0	0	0	1	1	0
Marseille	3	1	0	1	1	0	1	1	4	2	1
Qingdao	1	1	0	1	0	0	0	3	2	2	0
Reykjavik	1	1	0	1	1	0	0	1	1	2	0
Rio de Janeiro	2	1	0	1	0	0	1	1	1	4	0
Salvador Bahia	3	1	0	1	1	0	2	1	2	3	0
San Diego	2	1	0	1	1	0	1	3	1	2	0
San Francisco	2	1	0	1	1	0	1	3	2	4	1
Seattle	3	1	0	1	1	0	3	2	1	4	0
Sydney	2	1	0	1	1	0	0	2	1	4	0
Valencia	1	1	0	1	1	0	0	1	0	3	0
Varadero	1	1	0	1	0	0	0	2	0	1	0
Venice	1	1	0	1	0	0	0	1	0	6	0

Table 21. Beach Lover Performance table with colour gradation.

7.3 Culture freak profile

7.3.1 Description

These are a group of history students that love visiting museums and historical buildings. They consider essential *Museums* and *Monument buildings*; they also like *Cultural buildings, Religious buildings* and *Military buildings.*

GEOGRAPHICAL FEATURES			
Water landmarks (rivers, lakes,...)	*Indifferent*		
Coast	*Indifferent*		
Beaches	*Indifferent*		
Geographical landmarks	*Indifferent*		
Mountains	*Indifferent*		
HOSTING			
Campings	*Indifferent*		
Hotels	*Indifferent*		
Luxury hotels	*Indifferent*		
Residential buildings	*Indifferent*		
FACILITIES AND LEISURE			
Food stores	*Indifferent*		
Shopping	*Indifferent*		
Shops	*Indifferent*	Favourite shops	
Fairs	*Indifferent*		
Casinos	*Indifferent*		
Parks and gardens	*Indifferent*		
SPORTS			
Aquatic sports	*Indifferent*	Favourite aquatic sport	
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*Indifferent*		
Field sports	*Indifferent*	Favourite field sport	
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*Indifferent*	Favourite nature sport	
CULTURE			
Museums	*Essential*	Favourite museums	*Archeology museums*
Cultural buildings	*I like it*		
Monument buildings	*Essential*		
INTERESTING BUILDINGS			
Religious buildings	*I like it*		
Skyscrapers	*Indifferent*		
Industrial buildings	*Indifferent*		
Militar buildings	*I like it*		
Sport facilities	*Indifferent*		
Stadiums	*Indifferent*		

Figure 45. Culture freak Profile Interview

7.3.2 Criteria

The number of criteria that have to be evaluated is reduced to 6, only classes that have been evaluated by the DM in the preferences interview: *Cultural Building, Military Building, Monument Building, Museum, Religious Building* and *Archaeology Museum*. All of them have to be maximized.

Name	maximize / minimize
Cultural building	maximize
Military building	maximize
Monument building	maximize
Museums	maximize
Religious building	maximize
Archaeology museum	maximize

Table 22. Culture freak profile Criteria Set.

7.3.3 Alternatives

Cities that have zero tags in the groups *Museums* or *Monument building* are removed from the set of alternatives. As show in Figure 46 candidate cities have to accomplish at least one of the *Museums* subclasses or one of the *Monument buildings* subclasses. *Museums* is the class with more descendants: 28, thus the probability of having one of the *Museums* subclass is high. However, *Monument buildings* has only 4 subclasses, and this reduces considerably the probability of being in the alternatives set.

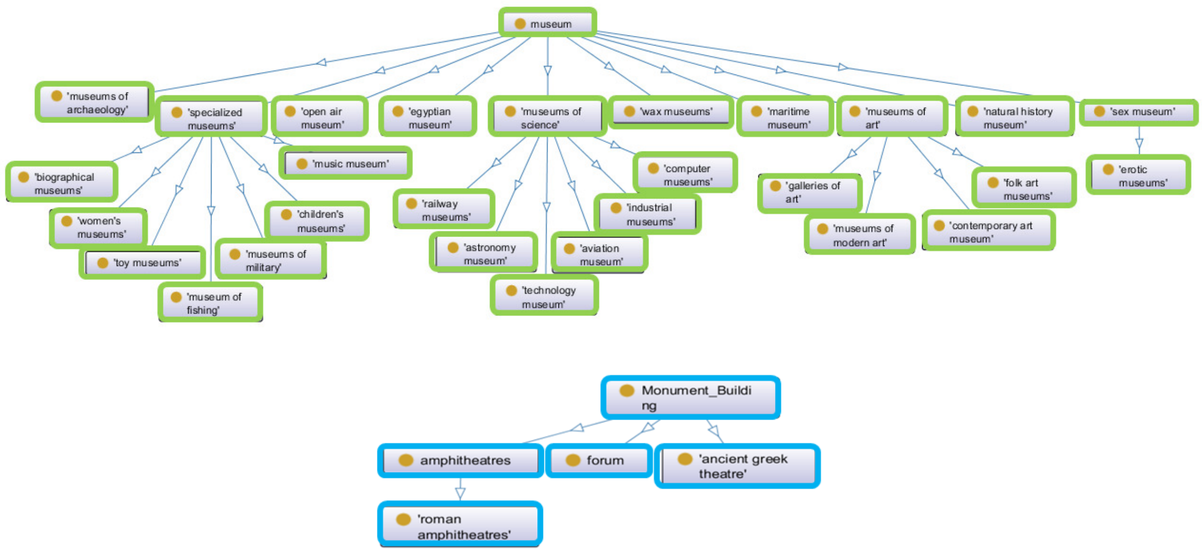


Figure 46. Essential classes of Culture Freak profile.

Our alternatives set is reduced to 22 cities, shown in Figure 47.

Figure 47. Set of alternatives for Culture Freak profile.

7.3.4 Performance table

In Table 23 we can see Performance Table generated for Culture Freak profile.

	Cultural Building	Military Building	Monument Building	Museum	Religious Building	Archaeology Museum
Barcelona	2	2	2	9	4	1
Bath Somerset	2	2	2	4	5	0
Birmingham	2	1	1	2	5	0
Bruges	2	0	1	2	5	0
Buenos Aires	2	2	1	1	3	0
Chester	2	1	1	4	4	0
Copenhagen	2	4	1	4	3	1
Hamburg	2	1	1	3	4	0
Heidelberg	0	5	1	1	4	0
Krakow	2	3	2	3	5	0
Kunming	1	1	1	3	2	0
Lisbon	2	2	1	4	7	1
Mexico City	2	3	1	4	4	0
Milan	2	1	1	2	3	0
Moscow	1	3	1	3	4	0
Paris	2	3	2	3	7	1
Rome	1	3	3	1	5	0
Saint Petersburg	2	3	1	6	3	0
Sao Paulo	2	3	1	2	2	0
Toronto	2	3	1	4	1	0
Turku	1	0	1	3	3	0
Zaragoza	1	1	1	1	5	0

Table 23. Culture Freak profile performance table.

7.3.5 Weights

In this test we have fewer criteria, thereupon weights are slightly higher than in other tests. In Table 24 we can see the criteria weights calculated for this test.

$$x = \frac{1}{numLeaves + (2 * numSuperclass) + (3 * numEssentialSuperclass)} = 0.0769307$$

where:

- *numLeaves = 1 (Archaeology museum).*
- *numSuperclass = 3 (Cultural building, Military building, Religious building).*
- *numEssentialSuperclass = 2 (Monument building, Museum).*

Criteria	Weight
Cultural building	0.15384615384615385
Military building	0.15384615384615385
Monument building	0.23076923076923078
Museum	0.23076923076923078
Religious building	0.15384615384615385
Archaeology museum	0.07692307692307693

Table 24. Culture Freak profile Weights

7.3.6 Thresholds

In Table 25 we can see the difference between thresholds calculated with method 1 (taking into account the otology definition) and with method 2 (taking into account max values of the alternatives set).

[redacted]

Table 25. Culture Freak profile thresholds

7.3.7 Touristic destinations ranking

In Figure 49 and Figure 48 we can observe the ranking obtained with Method 1 and Method 2. In this test results are quite similar. This is due to the fact that thresholds are only different for *Museum* and *Religious building*. With both methods Barcelona and Paris are the best cities.

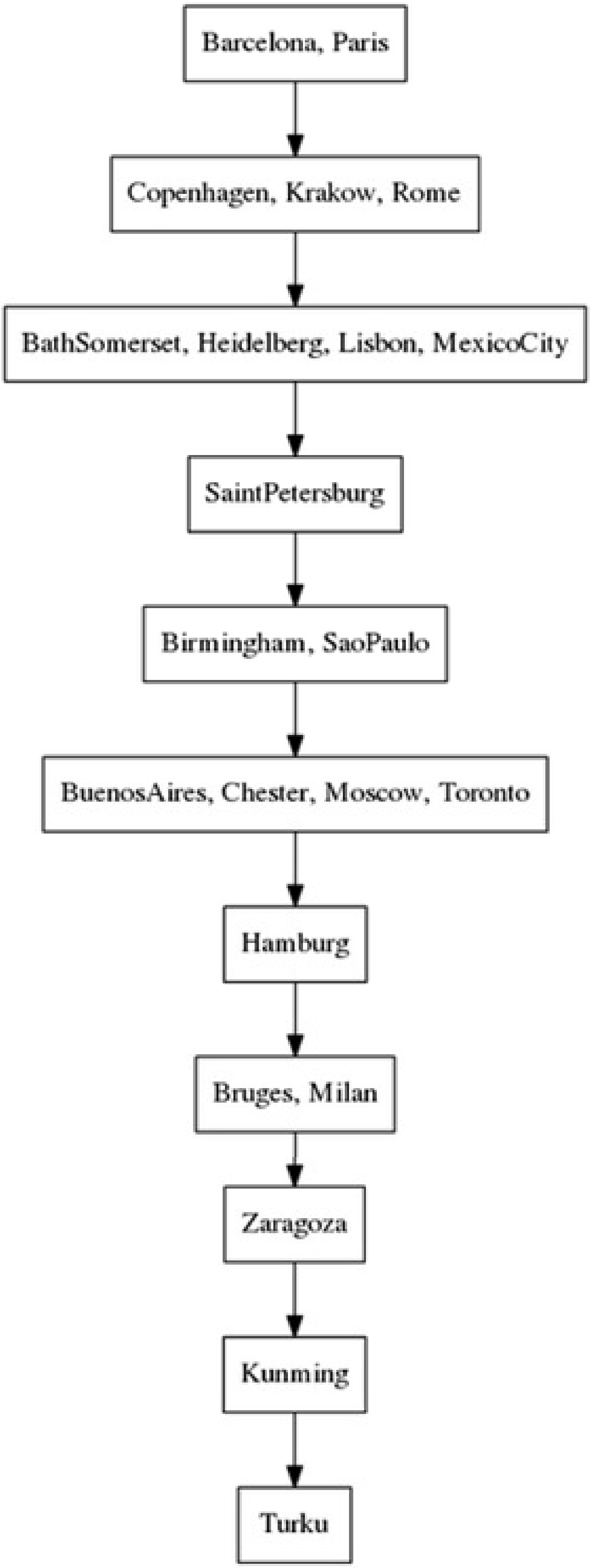


Figure 49. Culture freak city ranking with Method 1

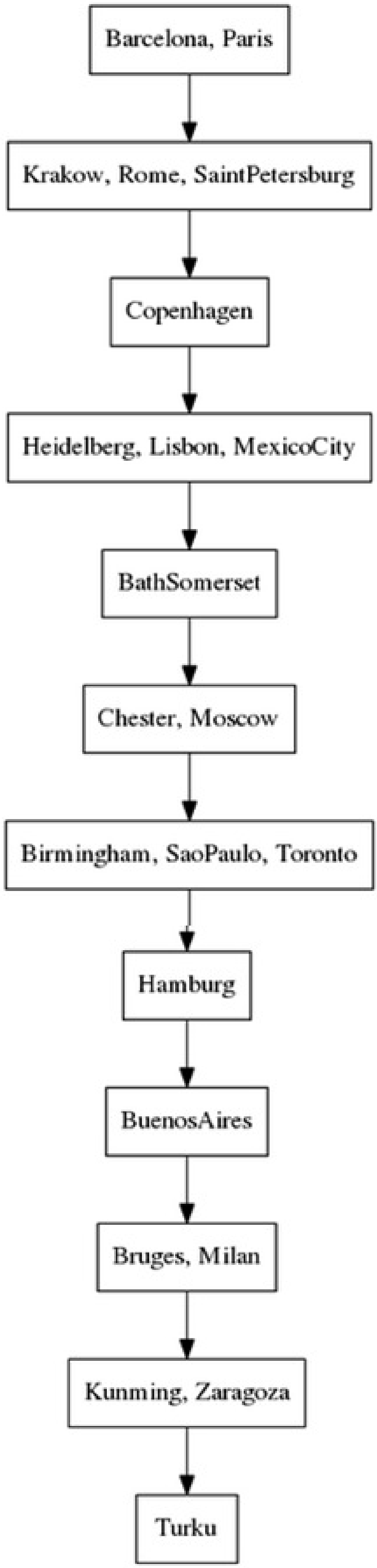


Figure 48. Culture freak city ranking with Method 2

	Cultural Building	Military Building	Monument Building	Museum	Religious Building	Archaeology Museum
Barcelona	2	2	2	9	4	1
Bath Somerset	2	2	2	4	5	0
Birmingham	2	1	1	2	5	0
Bruges	2	0	1	2	5	0
Buenos Aires	2	2	1	1	3	0
Chester	2	1	1	4	4	0
Copenhagen	2	4	1	4	3	1
Hamburg	2	1	1	3	4	0
Heidelberg	0	5	1	1	4	0
Krakow	2	3	2	3	5	0
Kunming	1	1	1	3	2	0
Lisbon	2	2	1	4	7	1
Mexico City	2	3	1	4	4	0
Milan	2	1	1	2	3	0
Moscow	1	3	1	3	4	0
Paris	2	3	2	3	7	1
Rome	1	3	3	1	5	0
Saint Petersburg	2	3	1	6	3	0
Sao Paulo	2	3	1	2	2	0
Toronto	2	3	1	4	1	0
Turku	1	0	1	3	3	0
Zaragoza	1	1	1	1	5	0

Table 26. Culture Freak Performance table with colour gradation.

7.4 Big family profile

7.4.1 Description

This is my family profile, we have three children aged five, eight and twelve. For us it is essential that the tourist destination has parks and gardens where children can play and have fun. We like doing nature excursions, thus *Geographical landmarks* are also essential. Moreover we like *Water landmarks*, *Museums* (for acquiring a bit of culture in travels) and *Nature sports.* My husband and my two boys love rugby and they always try to see a match when travelling, hence we also like *Field sports* and our favourite is *Rugby*. Finally, I have marked a feature I like: *Shops* being my favourite *Antiquarian.*

GEOGRAPHICAL FEATURES			
Water landmarks (rivers, lakes,...)	*I like it*		
Coast	*Indifferent*		
Beaches	*Indifferent*		
Geographical landmarks	*Essential*		
Mountains	*Indifferent*		
HOSTING			
Campings	*Indifferent*		
Hotels	*Indifferent*		
Luxury hotels	*Indifferent*		
Residential buildings	*Indifferent*		
FACILITIES AND LEISURE			
Food stores	*Indifferent*		
Shopping	*Indifferent*		
Shops	*I like it*	Favourite shops	*Antiquarian shops*
Fairs	*Indifferent*		
Casinos	*Indifferent*		
Parks and gardens	*Essential*		
SPORTS			
Aquatic sports	*Indifferent*	Favourite aquatic sport	
Martial arts	*Indifferent*	Favourite martial art	
Motor sports	*Indifferent*		
Ballet	*Indifferent*		
Field sports	*I like it*	Favourite field sport	*Rugby*
Skateboarding	*Indifferent*		
Popular running	*Indifferent*		
Nature sports	*I like it*	Favourite nature sport	
CULTURE			
Museums	*I like it*	Favourite museums	*Natural history museums*
Cultural buildings	*Indifferent*		
Monument buildings	*Indifferent*		
INTERESTING BUILDINGS			
Religious buildings	*Indifferent*		
Skyscrapers	*Indifferent*		
Industrial buildings	*Indifferent*		
Militar buildings	*Indifferent*		
Sport facilities	*Indifferent*		
Stadiums	*Indifferent*		

Figure 50. Big Family Profile Interview

7.4.2 Criteria

The number of criteria that have to be evaluated is ten. In this test we evaluate both criteria with highest number of descendants: *Field sport* (16) and *Museums* (28). All criteria have to be maximized.

Name	maximize / minimize
Field sport	maximize
Geographical landmark	maximize
Museum	maximize
Nature sport	maximize

Park	maximize
Shop	maximize
Water landmark	maximize
Rugby	maximize
Natural history museum	maximize
Antiquarian shop	maximize

Table 27. Big family profile Criteria Set.

7.4.3 Alternatives

Cities that have zero tags in the groups *Parks* or *Geographical landmark* are removed from the set of alternatives (see Figure 51).

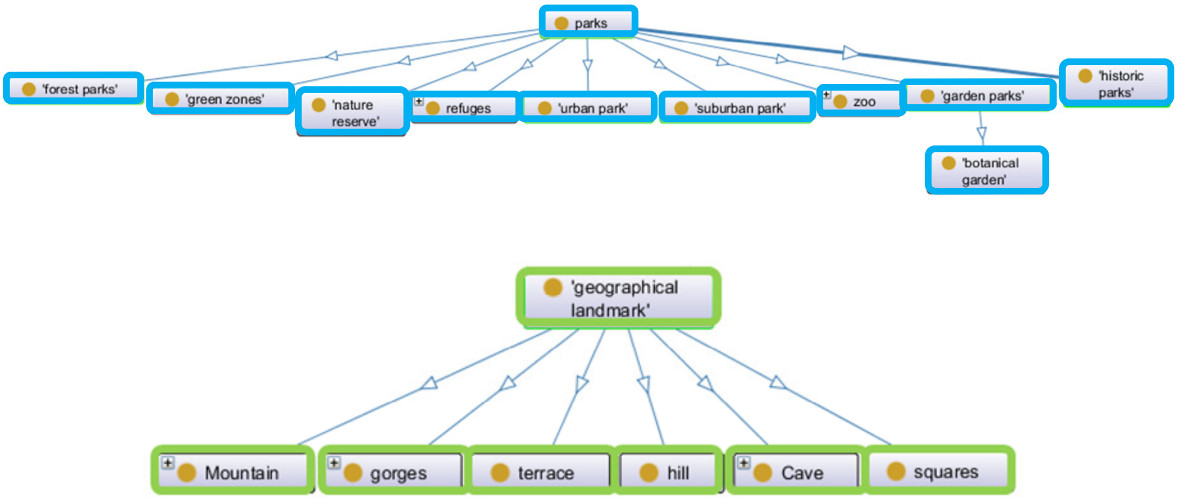


Figure 51. Essential classes of Big Family profile.

This is the test with highest number of alternatives: 52. Criterion market as essential have a large number of descendants, moreover *Parks* and *Geographical Landmarks* are features quite common in all cities description. Alternatives are shown in Figure 52.

Figure 52. Set of alternatives for Big Family profile.

7.4.4 Performance table

In Table 28 we can see Performance Table generated for Big Family profile.

	Field sport	Geographical landmark	Museum	Nature sport	Park	Shop	Water landmark	Rugby	Natural History museum	Antiquarian shop
Aberdeen	5	3	3	2	1	0	2	1	0	0
Amsterdam	3	2	5	1	2	1	5	0	1	0
Antwerp	2	2	3	1	1	0	3	0	0	0
Bath Somerset	10	3	4	1	2	1	4	1	0	0
Bilbao	2	3	2	2	1	1	3	0	0	0
Boston	3	2	4	1	2	0	2	0	0	0
Bristol	4	3	3	1	3	0	4	1	0	0
Bruges	1	1	2	1	1	0	3	0	0	0
Buenos Aires	7	1	1	1	3	1	2	1	0	0
Cambridge	4	1	0	1	2	0	1	1	0	0
Cape town	4	4	0	2	1	1	1	1	0	0
Cardiff	11	2	0	3	1	1	3	1	0	0
Chester	5	1	4	1	2	1	4	1	1	0
Chicago	3	2	7	1	4	1	5	0	1	0
Copenhagen	5	2	4	1	2	0	4	1	1	0
Dalian	2	3	1	1	1	0	3	0	0	0
Dubai	4	2	2	0	1	0	0	0	1	0
Glasgow	6	3	4	1	1	0	4	1	0	0
Granada	2	2	2	0	1	0	0	0	0	0
Hamburg	8	1	3	1	2	0	5	1	0	0
Hangzhou	0	2	0	1	2	0	5	0	0	0
Innsbruck	2	1	2	2	2	0	2	0	0	0
Istanbul	5	1	5	2	1	1	3	0	0	0
Kuala Lumpur	5	3	2	1	3	0	4	0	0	0
Kunming	9	5	3	1	2	0	4	0	1	0
Leeds	6	3	6	2	2	0	4	1	0	0
Lisbon	5	2	4	1	3	0	2	0	0	0
London	6	2	6	1	2	1	2	1	1	0
Madrid	2	1	3	1	1	0	2	0	0	0
Manchester	6	3	0	1	1	0	2	1	0	0
Marseille	3	5	6	1	1	0	2	1	1	0
Montreal	5	3	1	1	1	0	4	1	0	0
Moscow	6	2	3	1	4	1	2	1	1	0
Munich	2	3	8	1	1	1	2	0	1	0
Nanjing	2	2	3	1	2	0	3	0	1	0

Naples	3	4	4	0	2	0	0	1	0	0
New Delhi	0	2	0	0	1	0	0	0	0	0
Newcastle upon Tyne	4	4	3	1	1	1	4	1	1	0
Oslo	2	3	6	1	2	0	2	0	0	0
Reading Berkshire	5	1	2	3	1	0	4	1	0	0
Rio de Janeiro	3	3	4	1	1	1	4	0	1	0
Rome	6	2	1	1	2	0	2	1	0	0
Salvador Bahia	6	1	2	2	1	0	3	0	0	0
Salzburg	1	2	1	1	1	0	3	0	0	0
San Diego	6	3	6	1	3	0	2	1	1	0
San Francisco	4	2	5	1	3	1	4	0	1	0
Sao Paulo	6	2	2	1	1	1	2	1	0	0
Seattle	4	3	4	3	2	1	4	0	1	1
Shanghai	3	3	3	1	1	0	3	0	1	0
Singapore	9	2	1	1	3	0	3	1	0	0
Warsaw	4	4	5	3	4	0	3	0	1	0
York	2	3	6	1	1	1	4	1	1	0

Table 28. Big Family profile performance table.

7.4.5 Weights

In Table 29 we can see the criteria weights calculated for Big Family test.

$$x = \frac{1}{numLeaves + (2 * numSuperclass) + (3 * numEssentialSuperclass)} = 0.0526315$$

where:

- *numLeaves = 3 (Rugby, Natural history museum, Antiquarian shop).*
- *numSuperclass = 5 (Field sport, Museum, Nature sport, Shop, Water landmark).*
- *numEssentialSuperclass = 2 (Geographical landmark, Park).*

Criteria	Weight
Field sport	0.10526315789473684
Geographical landmark	0.15789473684210525
Museum	0.10526315789473684
Nature sport	0.10526315789473684
Park	0.15789473684210525
Shop	0.10526315789473684
Water landmark	0.10526315789473684
Rugby	0.05263157894736842
Natural history museum	0.05263157894736842
Antiquarian shop	0.05263157894736842

Table 29. Big Family profile Weights

7.4.6 Thresholds

In Table 30 we can see the difference between thresholds calculated with method 1 (taking into account the otology definition) and with method 2 (taking into account max values of the alternatives set).

						Method 1			Method 2		
Criteria	Is a leaf	Distance to leaf	Ontology #Descend.	Maxvalue all altern.	Maxvalue in test	Indifference Threshold	Preference Threshold	Veto Threshold	Indifference Threshold	Preference Threshold	Veto Threshold
Field sport	No	2	16	11	11	1	3		1	2	
Geographical landmark	No	1	6	5	5	0	1	2	0	1	2
Museum	No	2	28	10	8	2	5		1	2	
Nature sport	No	2	7	3	3	1	2		0	1	
Park	No	2	10	5	4	1	2	3	0	1	2
Shop	No	2	5	1	1	0	1		0	1	
Water landmark	No	2	8	6	5	1	2		0	1	
Rugby	Yes	0	0	1	1	0	0		0	0	
Natural history museum	Yes	0	0	1	1	0	0		0	0	
Antiquarian shop	Yes	0	0	1	1	0	0		0	0	

Table 30. Big family profile thresholds

7.4.7 Touristic destinations ranking

In Figure 54 and Figure 53 we can observe the ranking obtained with Method 1 and Method 2. In this test results are have more differences, proportionally to the number of differences in thresholds obtained with both methods.

The recommendation is very different in this case. Notice that the 4 best cities according to Method 1 are Bath Somerset, Kunming, Newcastle upon Tyne and Seatle, while Method 2 recommends Leads and Warsaw (which are in the second position in the ranking of Method 1). More surprisingly, Newcastle upon Tyne appears in the 4th position of the ranking of Method 2, Bath Somerset is in the 3rd position and the others in the second place.

The motivation for such differences in the preference order relies again on the thresholds. Method 1 is more tolerant than Method 2, because its values of the indifference and preference thresholds are larger in Method 1 than Method 2 in many criteria. Therefore, if we wish a city that clearly focuses on the criteria selected, we should go to Leads or Warsaw. If we want to open our possibilities to other kind of features (while prioritizing the selected criteria preferences), we could follow the recommendation of Method 1.

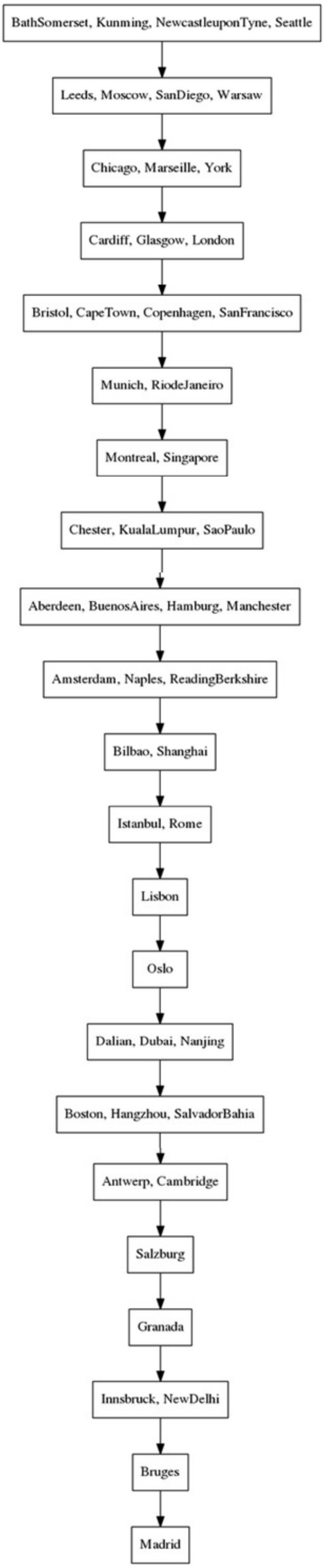


Figure 54. Big Family city ranking with Method 1

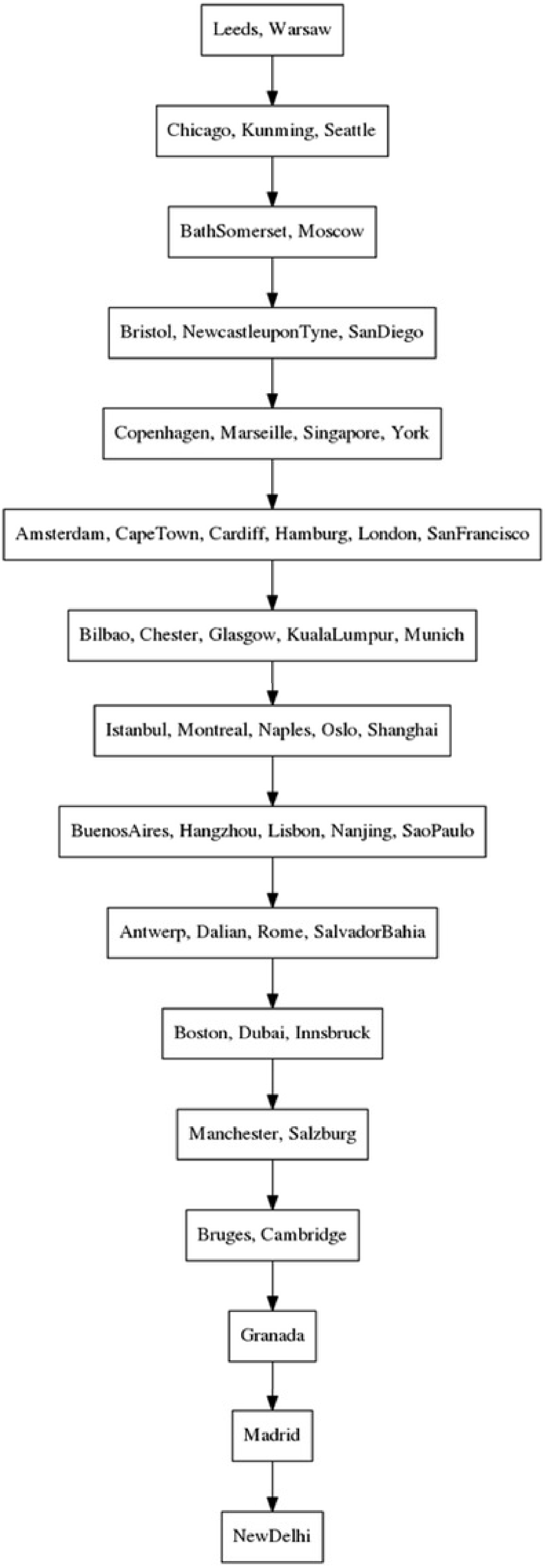


Figure 53. Big Family ranking with Method 2

	Field sport	Geographical landmark	Museum	Nature sport	Park	Shop	Water landmark	Rugby	Natural History museum	Antiquarian shop
Aberdeen	5	3	3	2	1	0	2	1	0	0
Amsterdam	3	2	5	1	2	1	5	0	1	0
Antwerp	2	2	3	1	1	0	3	0	0	0
Bath Somerset	10	3	4	1	2	1	4	1	0	0
Bilbao	2	3	2	2	1	1	3	0	0	0
Boston	3	2	4	1	2	0	2	0	0	0
Bristol	4	3	3	1	3	0	4	1	0	0
Bruges	1	1	2	1	1	0	3	0	0	0
Buenos Aires	7	1	1	1	3	1	2	1	0	0
Cambridge	4	1	0	1	2	0	1	1	0	0
Cape town	4	4	0	2	1	1	1	1	0	0
Cardiff	11	2	0	3	1	1	3	1	0	0
Chester	5	1	4	1	2	1	4	1	1	0
Chicago	3	2	7	1	4	1	5	0	1	0
Copenhagen	5	2	4	1	2	0	4	1	1	0
Dalian	2	3	1	1	1	0	3	0	0	0
Dubai	4	2	2	0	1	0	0	0	1	0
Glasgow	6	3	4	1	1	0	4	1	0	0
Granada	2	2	2	0	1	0	0	0	0	0
Hamburg	8	1	3	1	2	0	5	1	0	0
Hangzhou	0	2	0	1	2	0	5	0	0	0
Innsbruck	2	1	2	2	2	0	2	0	0	0
Istanbul	5	1	5	2	1	1	3	0	0	0
Kuala Lumpur	5	3	2	1	3	0	4	0	0	0
Kunming	9	5	3	1	2	0	4	0	1	0
Leeds	6	3	6	2	2	0	4	1	0	0
Lisbon	5	2	4	1	3	0	2	0	0	0
London	6	2	6	1	2	1	2	1	1	0
Madrid	2	1	3	1	1	0	2	0	0	0
Manchester	6	3	0	1	1	0	2	1	0	0
Marseille	3	5	6	1	1	0	2	1	1	0
Montreal	5	3	1	1	1	0	4	1	0	0
Moscow	6	2	3	1	4	1	2	1	1	0
Munich	2	3	8	1	1	1	2	0	1	0
Nanjing	2	2	3	1	2	0	3	0	1	0

Naples	3	4	4	0	2	0	0	1	0	0
New Delhi	0	2	0	0	1	0	0	0	0	0
Newcastle upon Tyne	4	4	3	1	1	1	4	1	1	0
Oslo	2	3	6	1	2	0	2	0	0	0
Reading Berkshire	5	1	2	3	1	0	4	1	0	0
Rio de Janeiro	3	3	4	1	1	1	4	0	1	0
Rome	6	2	1	1	2	0	2	1	0	0
Salvador Bahia	6	1	2	2	1	0	3	0	0	0
Salzburg	1	2	1	1	1	0	3	0	0	0
San Diego	6	3	6	1	3	0	2	1	1	0
San Francisco	4	2	5	1	3	1	4	0	1	0
Sao Paulo	6	2	2	1	1	1	2	1	0	0
Seattle	4	3	4	3	2	1	4	0	1	1
Shanghai	3	3	3	1	1	0	3	0	1	0
Singapore	9	2	1	1	3	0	3	1	0	0
Warsaw	4	4	5	3	4	0	3	0	1	0
York	2	3	6	1	1	1	4	1	1	0

Table 31. Big Family profile Performance table with colour gradation

7.5 Summary and discussion

We have tested OntElectre with real touristic data: DAMASK Ontology and DAMASK City matrix; and with four different profiles: Fashion Victim, Beach Lover, Culture Freak and Big Family. Results obtained in all tests are positive when contrasted manually regarding the input data (alternatives description). These results indicate that the methodology proposed is suitable for building recommender systems for tourists.

As we have also observed it is important that sources are updated. The different quality of information about cities features may cause that cities best informed have more chances to occupy the top positions in the ranking; this is the case of Cardiff. Mr.Mata already mentioned this lack of information in section "Future Work" of his Master Thesis: *"The information extraction developed in the first steps of the DAMASK project is not perfect and there is a high number of cities that have attributes with missing (represented by the ? symbol)"* [Mat12].

8 Conclusions and future work

In this master thesis, we have studied the problem of recommendation of objects described by a large set of linguistic tags. For the interpretation of the meaning of the tags, an ontology of reference is provided. In order to generate a ranking of these alternatives based on the preferences of the user, a complete new methodology has been developed, called OntElectre. In summary, this methodology has the following steps:

- The user profile is defined by selecting some key concepts of the ontology that will constitute the set of evaluation criteria. The user indicates his preference about these concepts.
- The list of tags of an alternative is automatically transformed in a numerical vector based on the criteria selected.
- The parameters of the criteria (i.e. the thresholds) are automatically calculated.
- The standard ELECTRE method may be then applied to obtain the ranking of the alternatives.

As has been proved in tests, OntElectre produces positive results with a low computing cost. An interesting improvement relative to the classical outranking methods is the automation of the thresholds calculation. Manually defining the thresholds is a tedious task for the user and it may be difficult to understand. Consequently, reducing the required user interaction in OntElectre, we increase its possibilities of being used.

Furthermore, we have developed two different methods to calculate thresholds: Method 1 (based in the ontology definition) and Method 2 (based in the alternatives performance). We could prove in tests that results obtained with both methods can be slightly different. Method 1, in general, is more tolerant than Method 2, because its indifference and preference thresholds tend to be larger than in Method 2.

Consequently, alternatives recommended with Method 1 comply with the user preferences but, this method opens the possibilities to alternatives with a medium performance in all criteria to be in top position. On the other hand, alternatives recommended with Method 2 clearly focus on the criteria selected by the user, prioritizing having highest performances in a criteria subset instead of having medium performances in all criteria set.

The selection of Method 1 or Method 2 depends on the concrete application of the system and on the quality of semantic information. When the quality of information of our alternatives is not uniform, we recommend Method 1.

As we have seen the proposed methodology is generic and can easily be adapted to any domain ontology in any field. OntElectre is highly parameterizable, in order to adapt it to another kind of decision problem it is only needed to change the ontology, the alternatives and the user profile file names. The only requisite is that these files respect the formats defined. Moreover, there is no restriction on the user profile nature; it may be implicit or explicit.

8.1 Future work

OntElectre constitutes a new approach to the management of semantic data in recommendation processes based on an outranking methodology. OntElectre first version main goal was to demonstrate the methodology designed achieved good results and performance. Now, that OntElectre goals have been proved, we can think about system enhancement adding complexity both to the process and to the data. Future work is therefore focussed on two complementary research lines:

- OntElectre enhancement
- Testing with Big Data

OntElectre enhancement may include the design and implementation of the following features:

- Management of preference values expressed as fuzzy sets with values in range [-1,2]: Current version of OntElectre works with a delimited set of integer values of preference: -1 (disliked), 0 (indifferent or unknown), 1 (liked) or 2 (essential). This delimited set of values facilitates the explicit introduction of preferences by the DM and the system testing. However, it could difficult the integration with implicit profiles, where preference values may be expressed in a range. This modification would imply a redefinition of the Performance Table calculation.

- Automation of the ontology simplification process: for case study I did a manual simplification of the ontology, removing all classes in the ontology that did not appear as alternatives' tags and were not ascendants of them. This process must be automated in order to deal with "bigger" ontologies.

- Design of a method to that automatically validates the proposed values of x, y, z, k, w (parameters used to obtain the thresholds)made in this work based on the analysis of the input data.

In the line of testing, we have proved that OntElectre performance is more than acceptable with an ontology and a set of alternatives we could consider of volume medium. Next steps imply testing OntElectre performance with big amounts of data.

8.2 Goals accomplished

To conduct this work, several goals were formulated and now it can be said that they have been completely accomplished and proved:

1. *Find a suitable model to represent the user's incomplete preferences on an ontology*: we have defined a user profile format that can be adapted to profiles obtained with different methods (explicit or hybrid). Moreover, OntElectre reduces considerably the user implication as thresholds are calculated automatically.

2. *Extend the classic ELECTRE method in order to exploit the preferences stored in the ontology for construction of a preference order:* our work has gone further than expected, as with the proposed methodology there is no need of modifying the ELECTRE method. The adaptations have been done in previous steps, consequently any existing tool implementing ELECTRE method could be used.

9 Annexes

I have included in this section some images and information that, although are not essential to understand the work, they allow the reader to further analysing.

9.1 DAMASK Ontology

In this section I show in detail the images corresponding to a reduced version of DAMASK Ontology. This reduced version corresponds to removing from DAMASK Ontology branches that do not appear in DAMASK city matrix as city tags. Mainly Geopolitical division descendants have been removed as these characteristics are not of touristic interest. In order to obtain the images I have used Protégée

Damask ontology is divided in four general city features:

- Geographical features
- Point of interest
- Activity
- Geopolitical division

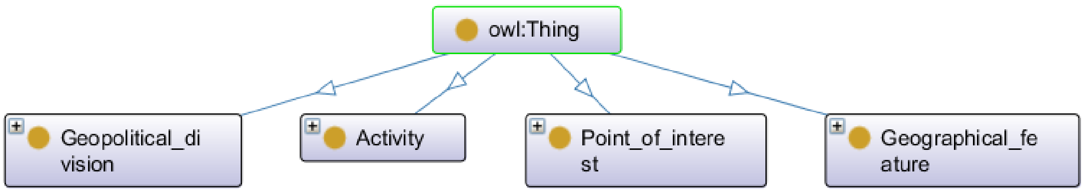


Figure 55. DAMASK Ontology - Principal features

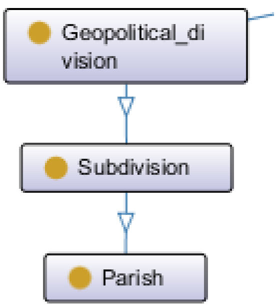


Figure 56. DAMASK Ontology - Geopolitical_division

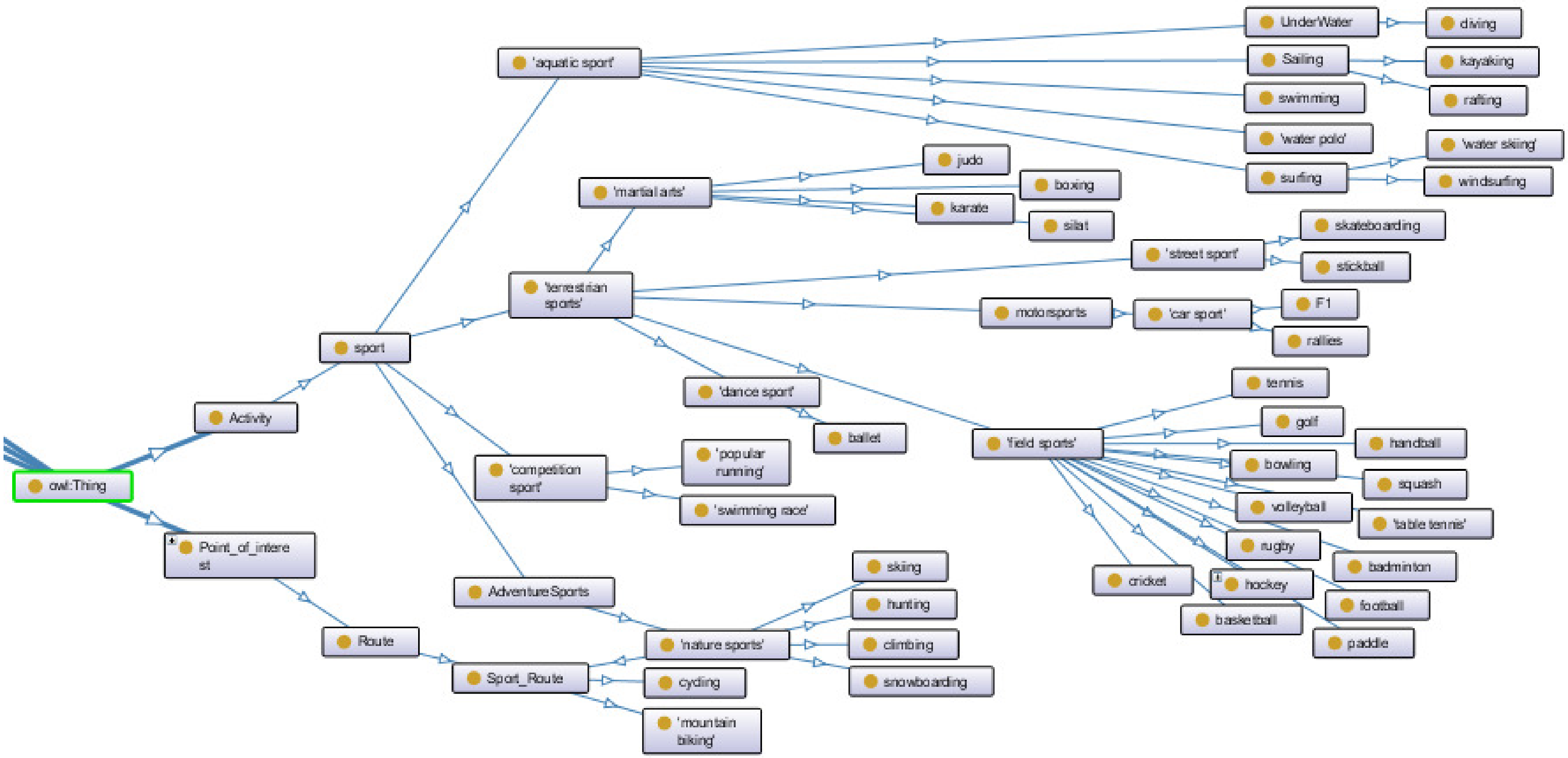


Figure 57. DAMASK Ontology - Activity -> Sport

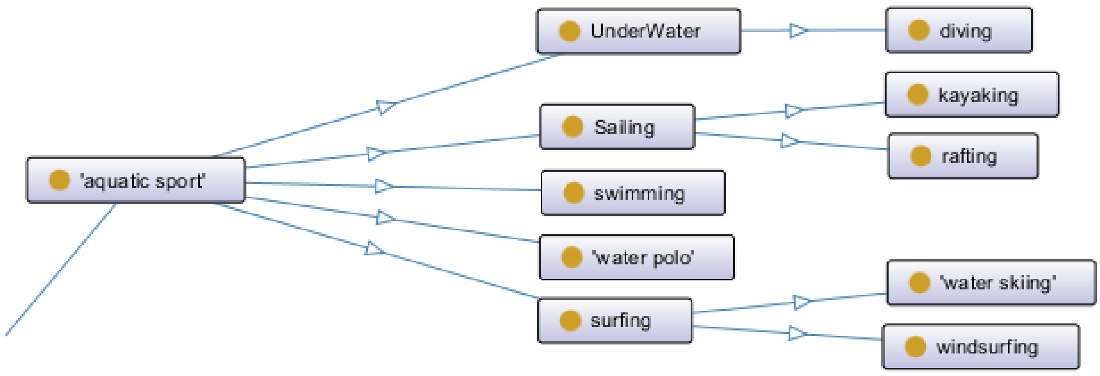


Figure 58. DAMASK Ontology - Activity -> Sport -> Aquatic Sport

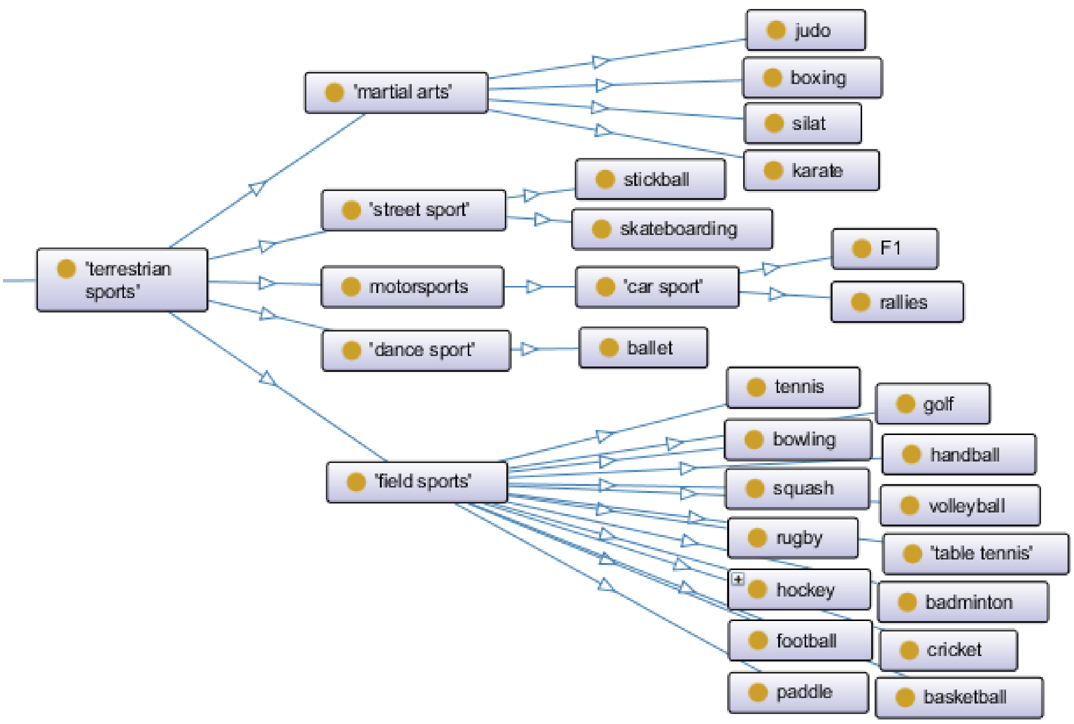


Figure 59. DAMASK Ontology - Activity -> Sport -> Terrestrian Sport

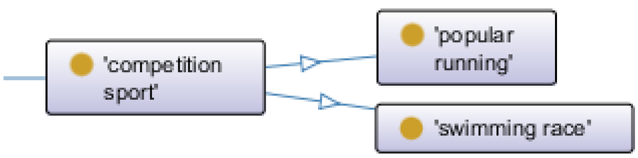


Figure 60. DAMASK Ontology - Activity -> Sport -> Competition Sport

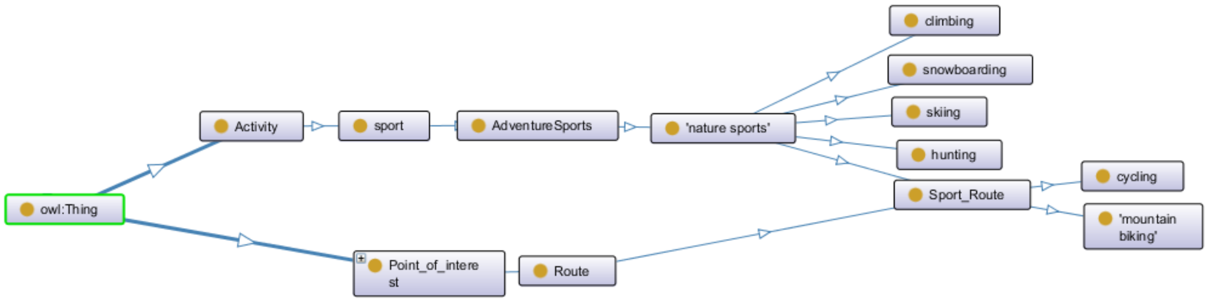


Figure 61. DAMASK Ontology - Activity -> Sport -> Adventure Sports

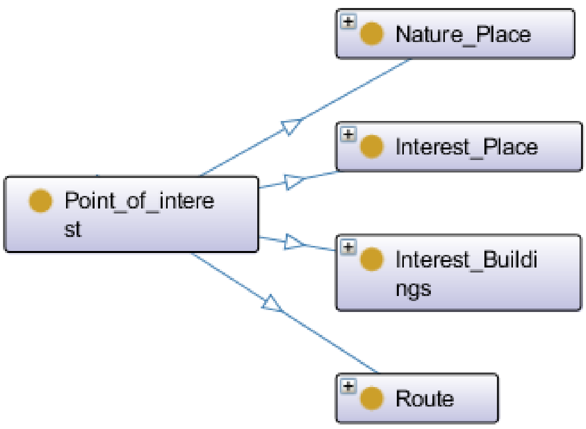


Figure 62. DAMASK Ontology - Point of interest

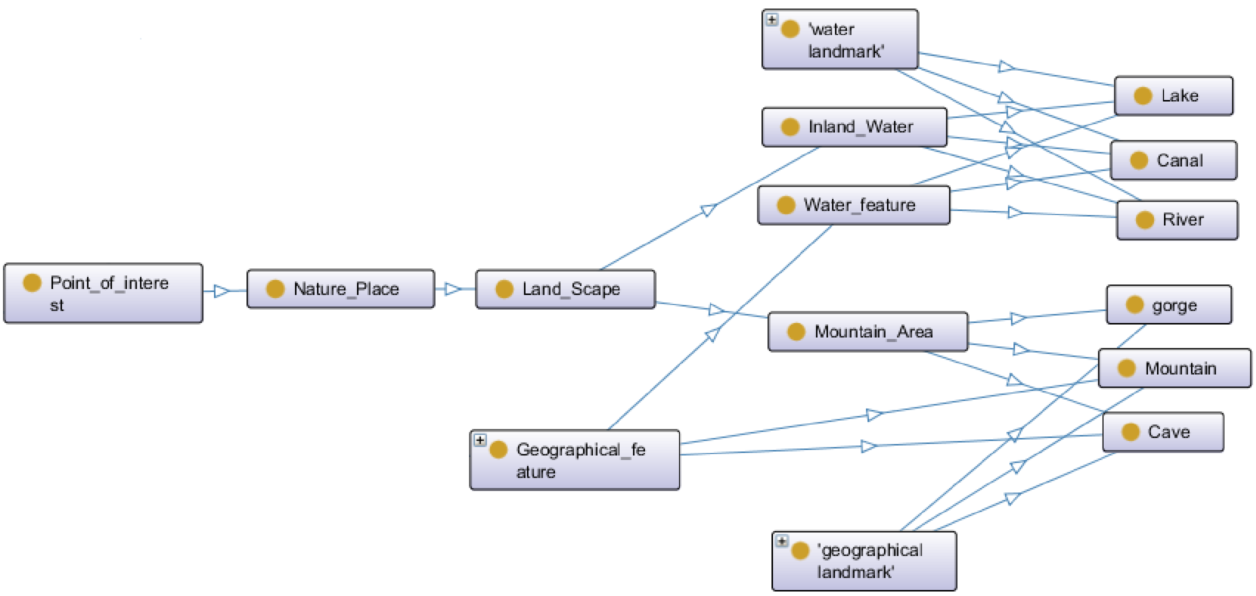


Figure 63. DAMASK Ontology - Point of interest -> Nature place

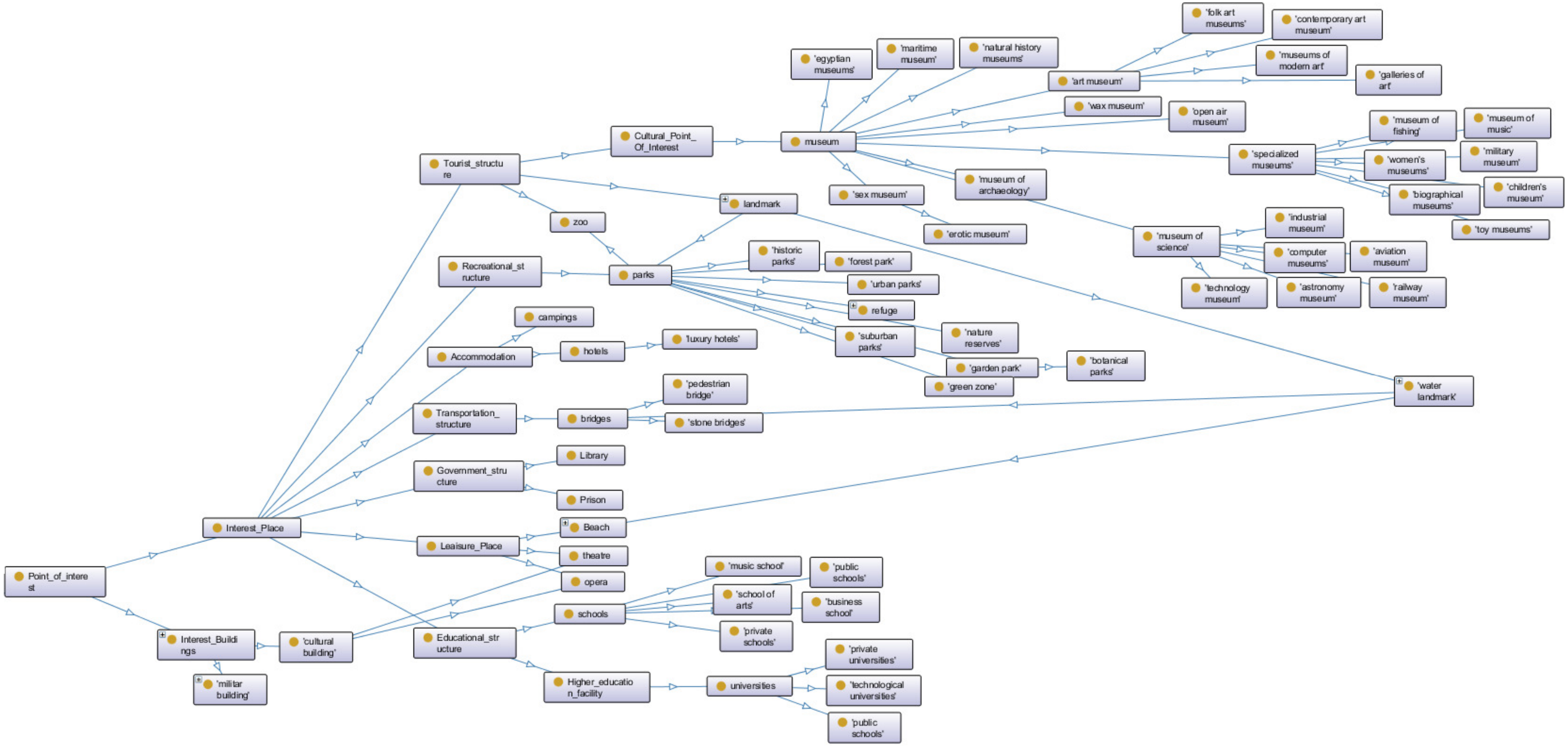


Figure 64. DAMASK Ontology - Point of interest -> Interest place

Ontology-based construction of outranking relations for decision support

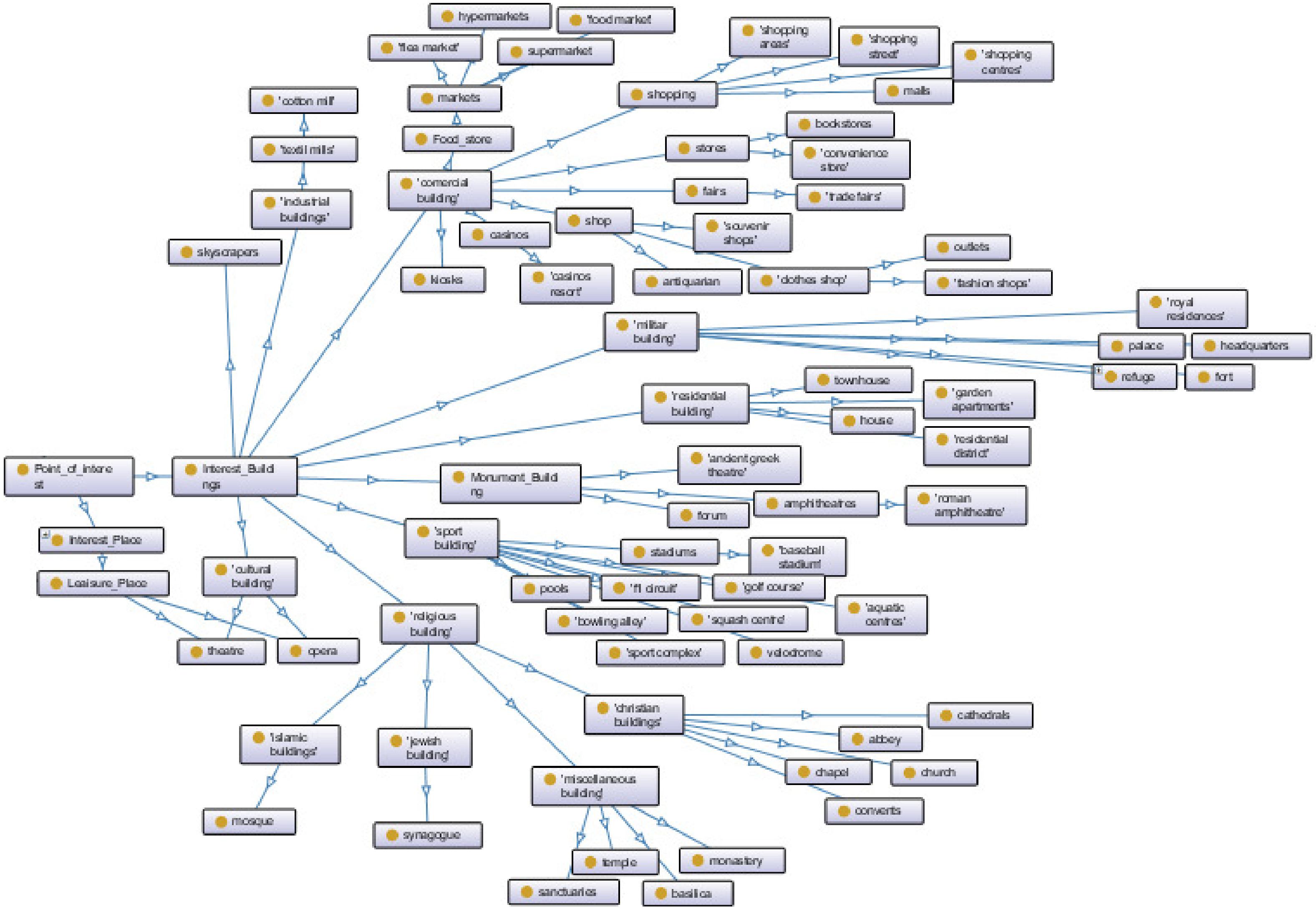


Figure 65. DAMASK Ontology - Point of interest -> Interest buildings

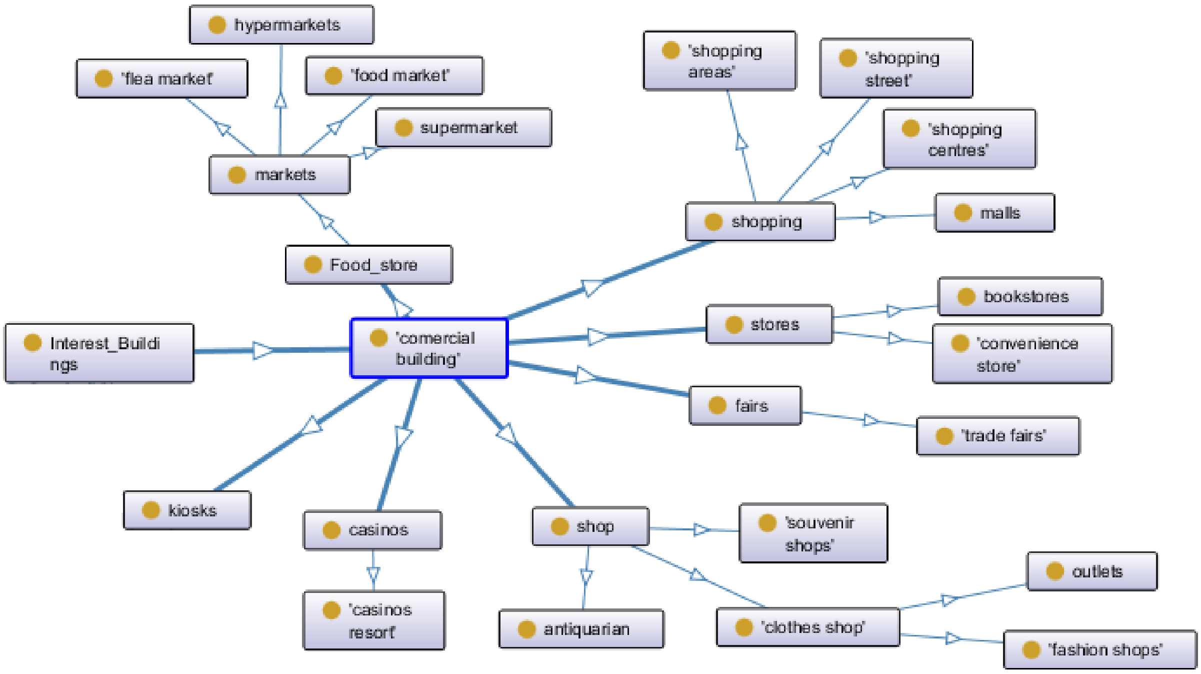


Figure 66. DAMASK Ontology - Point of interest -> Interest buildings -> Commercial Building

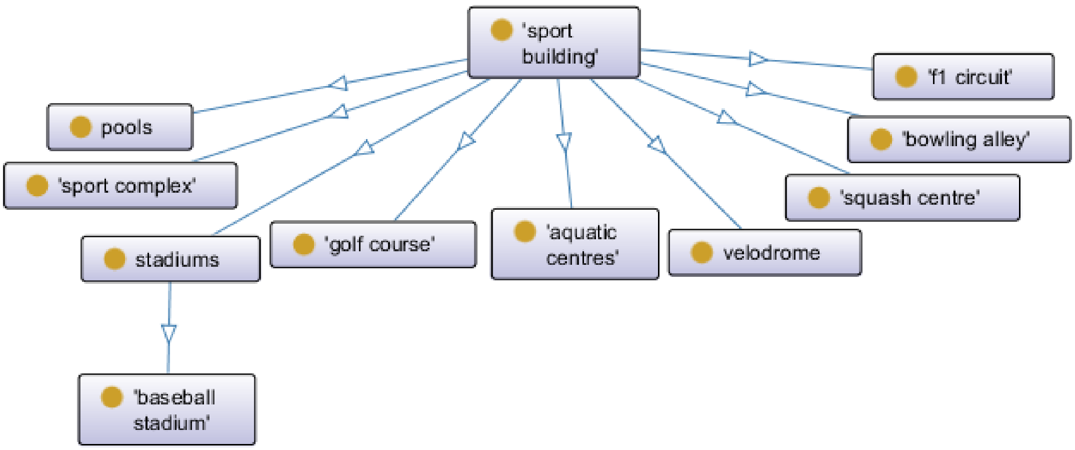


Figure 67. DAMASK Ontology - Point of interest -> Interest buildings -> Sport building

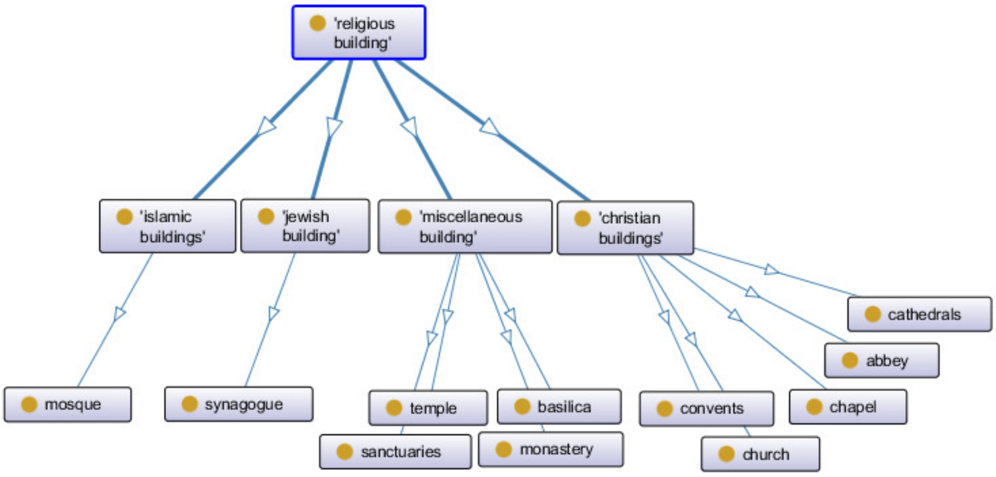


Figure 68. DAMASK Ontology - Point of interest -> Interest buildings -> Religious building

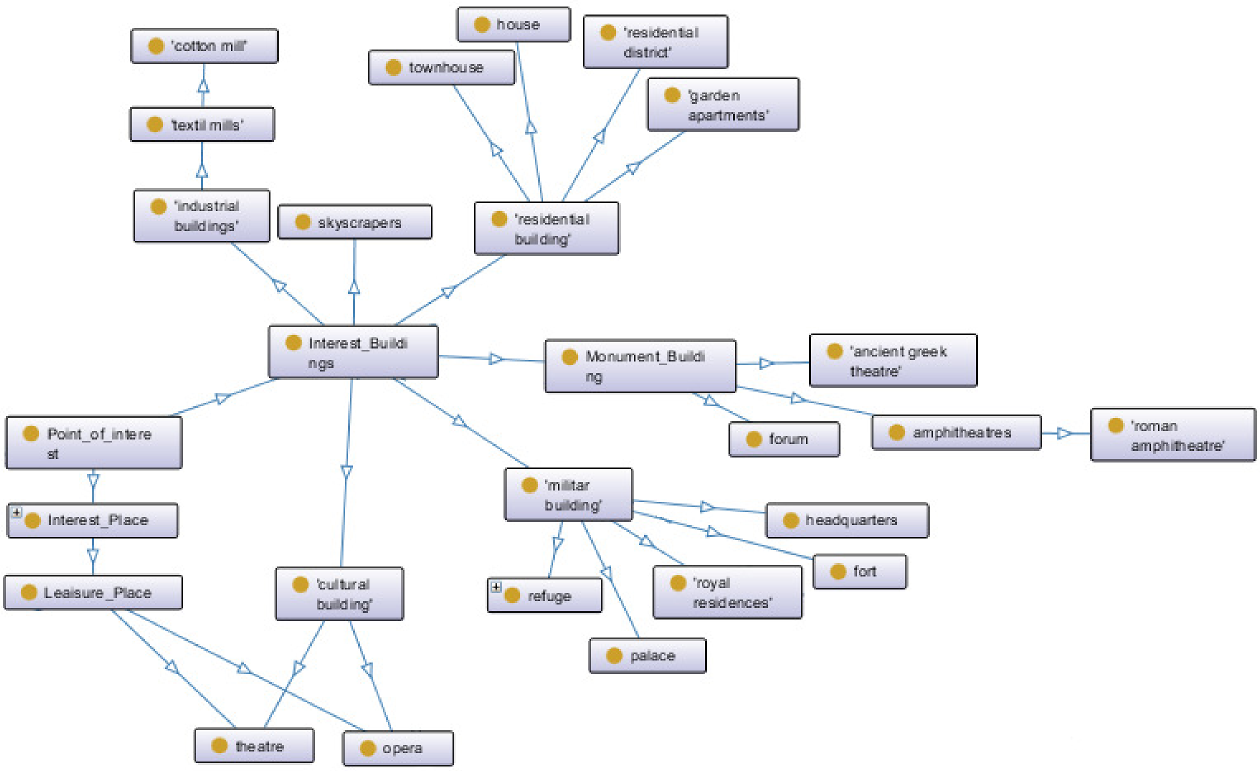


Figure 69. DAMASK Ontology - Point of interest -> Interest buildings

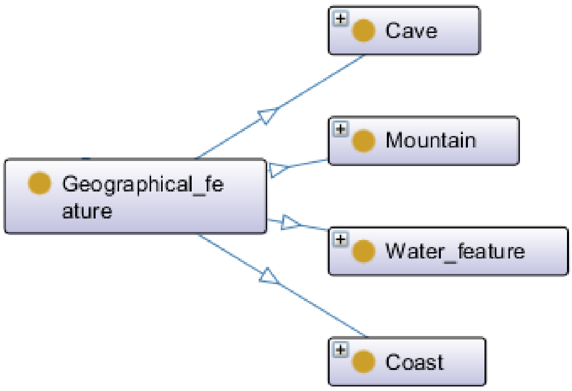


Figure 70. DAMASK Ontology - Geographical feature (I)

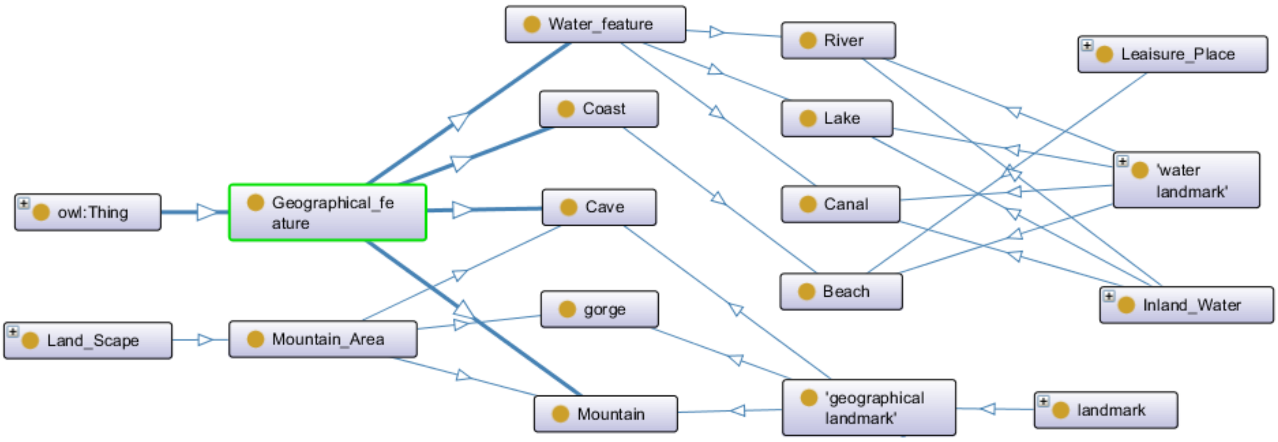


Figure 71. DAMASK Ontology - DAMASK Ontology - Geographical feature (II)

10 References

Books:

[Ish13] Alessio Ishizaka, Philippe Nemery, "Multi-Criteria Decision Analysis. Methods and software", John Wiley & Sons, Ltd 2013. ISBN: 978-1-119-97407-9

[Arp15] Robert Arp, Barry Smith, Andrew D.Spear, "Building Ontologies with Basic Formal Ontology", MIT Press, 31 jul. 2015 ISBN:978-0-262-52781-1

[Val13] Aida Valls, Antonio Moreno and Joan Borràs, " Preference representation with ontologies, in book: Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications ", Editors: M. Doumpos, E.Grigoroudis, Publisher: John Wiley & Sons, pp. 77-99, 2013. ISBN: 978-1-1199-7639-4

[Fen01] Dieter Fensel, "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce", Springer Berlin Heidelberg 2001

[Gre05] Greco, S., Figueira, J., Ehrgott, M., (EDs), “Multiple criteria decision analysis. State of the art Surveys”, Springer, 2005.

Papers in journals:

[Roy68] Roy, Bernard (1968). "Classement et choix en présence de points de vue multiples (la méthode ELECTRE)". La Revue d'Informatique et de Recherche Opérationelle (RIRO) (8): 57–75

[Roy74] Roy, B., "Critères multiples et modélisation des préferénces: l’apport des relations de surclassment", Revue d’Economie Politique, 1, 1–44. (1974)

[Sas13] Ana Sasa Bastinos, Marjan Krisper, "Multi-criteria decision making in ontologies", Information Sciences, Vol.222, 10 February 2013, Pages 593-610 (2013)

[Cal13] Silvia Calegari, Gabriella Pasi, "Personal ontologies: Generation of user profiles based on the YAGO ontology", Information Processing and Management 49 (2013) 640-658

[Bla11] Yolanda Blanco-Fernández, Martín López-Nores, José J.Pazos-Arias, Jorge García-Duque, "An improvement for semantics-based recommender systems grounded on attaching temporal information to ontologies and user profiles", Engineering Applications of Artificial Intelligence 24(2011) 1385-1397

[Jun13] Jungmin Kim, Hyunsook Chung and Gukboh Kim, "TV Program Searching and Ranking for Supporting TV Personalization", International Journal of Multimedia and Ubiquitous Engineering Vol. 8, No. 1, January, 2013

[Ely14] Yasyn ELYUSUFI, Hamid SEGHIOUER, Mohammed Amine ALIMAM, "Building Profiles Based on Ontology for Recommendation Custom Interfaces", 978-1-4799-3824-7/14 ©2014 IEEE

[Ali14] Mohammed Amine ALIMAM, Hamid SEGHIOUER, Yasyn ELYUSUFI, "Building profiles based on ontology for career recommendation in E-leaming context.", 978-1-4799-3824-7/14 ©2014 IEEE

[Oua15] Sara OUAFTOUH, Ahmed ZELLOU, Ali IDRI, "User profile model: a user dimension based classification", 78-1-4799-7560-0/15/$31 ©2015 IEEE

[Suc07] Suchanek, F.M. Kasneci, G. and Weikum, G. "YAGO: A core of semantic knowledge", WWW (pp.697-706) 2007

[Suc08] Suchanek, F.M. Kasneci, G. and Weikum, G."YAGO: A large ontology from wikipedia and wordnet", Journal of Web Semantic 6(3), 203-217 2008

[Ash15] Jamshaid Ashraf, Elizabeth Chang, Omar Khadeer Hussain, Farookh Khadeer Hussain, "Ontology usage analysis in the ontology lifecycle: A state-of-the-art-review", Knowledge-Based Systems 80 (2015) 34-37, March 2015

[Per92] Perny, P., Roy, B., “The use of fuzzy outranking relations in preference modelling”, Fuzzy Sets and Systems, vol. 49, pp.33-53, 1992.

Papers in Conferences:

[Bor12] Joan Borràs, Aïda Valls, Antonio Moreno, David Isern, "Ontology-based management of uncertain preferences in user profiles", International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2012: Advances in Computational Intelligence pp 127-136.

[Mar16] Miriam Martínez-García, Aïda Valls, Antonio Moreno, "Construction of an outranking relation based on semantic criteria with ELECTRE-III", International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2016: Information Processing and Management of Uncertainty in Knowledge-Based Systems pp 238-249.

[Vic11] C. Vicient, D. Sánchez, and A. Moreno, "Ontology-Based Feature Extraction," in Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, NLPOE, Lyon, France, 2011, pp. 189-192.

[Pin07] Pingfeng Liu, Guihua Nie, Donglin Chen, "Exploiting Semantic Descriptions of Products and User Profiles for Recommender Systems ", Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

[Nau08] Yannick Naudet, Armen Aghasaryan, Yann Toms, and Christophe Senot, "An Ontology-based Profiling and Recommending System for Mobile TV", 2008 Third International Workshop on Semantic Media Adaptation and Personalization, 978-0-7695-3444-2/08 2008 IEEE DOI 10.1109/SMAP.2008.9

[Bat12] Brahim Batouche, Damien Nicolas, Hedi Ayed, Djamel Khadraoui, "Recommendation of Travelling Plan for Elderly People According to their Abilities and Preferences", 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), 978-1-4673-4794-5/12/$31.00 c 2012 IEEE

[Ajm13] Stuti Ajmani, Hiranmay Ghosh, Anupama Mallik, Santanu Chaudhury, "An ontology based personalized garment recommendation system" , 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)

[Bor14] Joan Borràs, Antonio Moreno, Aida Valls, "Intelligent tourism recommender systems: A survey", Expert Systems with Applications 41 (2014) 7370-7389

[Bre07] K. Breitman, M.Perazolo, "Using Formal Ontology Representation and Alignment Strategies to Enhance Resource Integration in Multi Vendor Autonomic Environments", Engineering of Autonomic and Autonomous Systems, 2007 EASe'07. Fourth IEEE International Workshop

[Koo05] Karin Koogan Breitman, Carolina Howard Felicíssimo, Marco Antonio Casanova, "CATO - A Lightweight Ontology Alignment Tool, CAISE 2005

[Vas12] Del Vasto-Terrientes, L., Valls, A., Zielniewicz, P., Slowinski, R., "Extending Concordance and Discordance Relations to Hierarchical Sets of Criteria in ELECTRE-III Method", MDAI (Modeling Decisions on Artificial Intelligence), Girona, November 2012.

Ph.D. and Master Thesis:

[Bor15] Joan Borràs Nogués "Semantic Recommender Systems", Universitat Rovira i Virgili, 2015

[Mat12] Ferran Mata Arcas "Obtaining general concepts that represent a set of objects using ontologies", Universitat Rovira i Virgili, 2012

[Vic09] Vicient, C. 2009. "Extracció basada en ontologies d 'informació de destinacions turístiques a partir de la Wikipedia". Universitat Rovira i Virgili. Final Year Project, URV-ETSE.

[Vic10] Vicient, C. 2010. "Ontology-based Information Extraction". Master Thesis, URV-ETSE.

Others:

[Mor11] Vicient, C. & Moreno, A. 2011. "Data-Mining Algorithms with Semantic Knowledge" DAMASK Internal project report T3.1 Damask Ontology. Project Deliverable, URV.

[Bre07] Bremner, C. 2007. "Top 150 City Destinations: London Leads the Way". Euromonitor International. http://blog.euromonitor.com/2007/10/top-150-city-destinations-london-leads-the-way.html

URLs:

Protégé: http://protege.stanford.edu/

Apache Jena: https://jena.apache.org/

Diviz: http://www.decision-deck.org/diviz