Chapter 1

Intelligent Information Agents

1.1 Introduction to Agents

Intelligent agents are software entities that support autonomous, deliberative, proactive and goal-directed behaviour and can socially interact with other intelligent agents to form a multi-agent system. Each agent is capable of independent (or autonomous) action, acting on behalf of an owner that may have different goals and motivations, to satisfy user goals (goal oriented), rather than constantly reacting to what they are told. In this chapter, the software agent paradigm and its possible application to the Internet and information retrieval on the WWW will be explained. Many papers describe agents in a very anthropological way, with terms and attributes often used to describe humans (like intelligence, autonomy, learning, communication, etc.).

One of the reasons for this may be to suggest some kind of similarity between agents and humans. Consequently, these terms should be read and understood with caution, they are used here for reasons of familiarity and their intention is not to suggest that currently existing agents exhibit anything near human-like qualities. The properties of a software agent system, the tasks of what it does and how it does that, can be described from two points of view, a psychological and a technical[86].

The first is an abstract way of describing an agent, from the point of view of
the user to whom the underlying techniques are transparent. As a second point, even though the exact techniques, mechanisms and algorithms used do not constitute part of a definition of agent hood, they are the means to implement the described properties. Looking at existing projects to see how they actually approach the implementation of agent-like behaviour is certainly of interest this will not necessarily result in a better understanding or definition of what an agent is or should be, but will hopefully bring in some new leads and ideas of how the ideal agent could be accomplished.

1.1.1 An agent from the users point of view

Let us define Agent by taking one a classical example, the human travel agent this agent is an expert that specializes in a knowledge domain (travels) and offers this as a service. He is called up once and then he works autonomously and most likely returns with a result a while later. Maybe the query for a travel connection has to be refined thus the agent communicates with the user and other information sources the agent uses his intelligence to reason and reach the goal (finding the right travel connection), while having other goals with different priorities at the same time. If the user is a frequent customer, the agent will most likely learn some of the users preferences and sometimes pro-actively suggest or assume something without explicitly being asked for it.

The main emphasis here is on the fact that the agent alleviates the user of some work, because a task can be delegated to it. On a lower level (with a simple agent) the user saves time, because he does not have to do it himself even if he could. On a higher level (with a powerful and very capable agent), the user gains some extra possibilities, because he lacks the domain-specific knowledge to do the task himself. Maybe the agent does not employ much intelligence to satisfy a query (e.g. Just passing it on to a travel database), but that remains transparent to the user. How the agent carries out its task
internally does not matter, as long as it comes up with a qualified answer to the users need and thus provides a solution to the problem. A related issue is the interface that is used to communicate with agents. Naturally it should be intuitive and easy-to-use (maybe natural language typed into a keyboard or even spoken), but such a definition is very ambiguous.

Humans tend to characterize complex systems like human beings, describing their behaviour using attitudes like knowledge, belief, intention and obligation. Thus the agent metaphor may improve the way a user interacts with an agent. Takes the approach of defining agents through mentalist notions one step further, by requiring that agents should not only be describable with such human attitudes, but that they actually need to be based on a formal logical framework that consists of such attitudes and allows to reason about them. Attitudes are divided into two groups. Information attitudes store the agents knowledge and pro-attitudes store its actions. So the human attitudes are not simulated, but the core of the agent is actually modeled using these attitudes and a formal logic to represent and manipulate them. However, this view of agents is already an implementation-related issue which rules out many other agent models and is therefore too narrow for this discussion.

Mentions two broad problems to be solved with respect to the user, competence and trust. Competence is the knowledge the agent needs to decide when to help the user, what to help the user with and how to help the user and trust exists when the user feels comfortable delegating tasks to the agent the acceptance depends mostly on the solution of these two problems; on the other hand such a quality requirement is true for practically every kind of software.

The issues discussed here also apply to an information retrieval agent that finds information for the user on the WWW the agent should be a helpful tool that the user can delegate this task to, working autonomously by itself. It should employ intelligence, its specific domain knowledge, reasoning
and learning to search and find the information the user wants the agent should fulfill the users information needs both reactively in response to a request, but also pro-actively suggesting new information that it deems interesting. A perfect IR agent would interface with the user in an intuitive way, maybe understanding the users natural language.

1.1.1.1 Agent properties

The following part will list some of the properties that are frequently mentioned, while not claiming to be complete or extensive. As already noted by [22] and [69], agents act, they do something the task they accomplish or the service they offer is their most basic property.

1.1.1.2 Environment

The environment the agents live (sense and act) in determines their scope, they can sense their surroundings (input) and affect them through their actions (output)[22]. A frequently mentioned part of the agent paradigm is that agents live in a real environment. In this case real does not have the meaning of physically real as used in connection with robots, because the agents are software agents. This usually results from the fact that the environment is open (very large or virtually unlimited in size) and inhabited by other agents or entities that may act in an undeterminable way the Internet and the WWW share these qualities, it is impossible for an agent to know in advance what it will encounter. Agents should therefore be fault-tolerant and have a certain robustness. Otherwise, users or agents they interact with in their environment may do damage to them (on purpose or unintentionally). Just the same, an agent should not be the cause of problems or damage to its environment.
1.1.1.3 Intelligence

Basically there are two approaches in AI to model intelligence: symbolic knowledge-based systems and reactive architectures. In a deliberative symbolic approach, a symbolic model of the environment and a set of action descriptions are created; the agent then uses symbolic reasoning to combine a sequence of such actions to a plan. The plan describes a path of action that leads to a predicted desired goal state, which the agent is deliberately trying to reach.

Reactive systems are built on the basic assumption that intelligent behaviour can be generated without explicit representation of the world model, but rather emerging from a complex system of behaviours. Essentially there is a set of rules or behaviours from which one is chosen or activated according to the currently sensed environment.

Since both of these approaches are imperfect, hybrid systems have been developed in an attempt to combine the advantages of both. Some properties that are still problematic or difficult to implement are dynamic generation of goals (usually these are predefined and fixed) and dealing with multiple, possibly conflicting goals. A good overview about this topic can be found in [97].

For example, if an URL is sent to a WWW server to retrieve a document, it will either return that document or not (because of a time-out or if it does not exist). Apparently an action like that is more like a function call and can very simply be used by an algorithm. Whether such an algorithm can still be called intelligent or not is a very debated issue that will not be discussed here.
1.1.1.4 Learning

Learning is related to or sometimes seen as part of intelligence, it can also be described as adaption with the goal of improving or optimizing the own performance[34]. Single agents can learn to improve their knowledge or their problem-solving method. Within a multi-agent system the improvement can also aim towards the interaction or cooperation the reward function reflects this in a way that agents try to maximize the combined reward of the whole system. In a competitive environment however, agents would usually behave selfish, trying to improve only themselves and maximize their own reward.

Another distinction can be made with respect to the systems architecture. Either it is fixed and designed exactly to perform its task, in which case learning takes place through knowledge data being modified or the architecture itself is adapted and evolves as part of the learning process.

1.1.1.5 Autonomy

Autonomy is sometimes used to explain that the agent does something without constant user interaction or supervision, or even more detached, without direct human intervention at all. But this is a slightly inaccurate description, as it does include simple agents who only act in response to a users query therefore the emphasis is also on an agent having its own agenda or goals to pursue. It observes its environment to recognize changes and takes action when a certain state or change is observed. Such a behaviour can be seen as autonomy, since the agent itself can decide when to do what, it is not following a direct external order this implies intelligence to a certain degree, some kind of rule set that the agent uses to make these decisions.

In a multi-agent system or an environment with many agents, autonomy also means being separate from other agents. Each of the agents can be clearly distinguished from the others, it is an encapsulated entity, it has its own
internal state and goals which may be different or even contradictory from those of other agents. Note that this does not rule out a cooperative multi-agent system where some agents depend on others (voluntarily or not), because they still have their own goals and are thus autonomous, yet they may not be independent the agent maintains the distinction between itself and its environment.

1.1.1.6 Communication

Agents should be able to interact - with other agents and the user (either directly or via an interface). To be able to communicate, agents need a standardized communication language whose syntax and semantics are clearly defined. Both procedural and declarative languages have advantages. Depending on the underlying agent model and implementation either one will be better suited to communicate the agents expressions. Agent implementations based on a speech act, ontology or belief model tend to use a declarative language to express their speech. By virtue of being sent, a message is intended to result in some action being performed - see [21],[43] and [25]. Contrary to that, a procedural language seems to be useful for mobile agent approaches to send programs or blocks of code to another location [60] for an example). In any case, communication is one of the important parts of software agents, since they interact with their environment mainly by sending messages or calling functions in one way or another.

The communication in a community of many agents can be direct, with each agent talking to each other this requires some self-organizing architecture, each agent has to know about the other agents presence and capabilities. It can also be indirect like in a blackboard system, via a message router or a mixed approach in which communication is assisted by specialized agents (e.g. by an agent name server or service broker). With respect to autonomy, each agent must be able to initiate messages [69]. So the communication must be based on peer-to-peer connections, which rules out client-server based protocols and architectures.
1.1.2 Multi-agent Systems

As already discussed in the communication property, many approaches consist of multiple agents two main types may be distinguished, based on competition or cooperation. In a competitive architecture the different agents offer solutions to the same problem and compete with each other to offer a better solution than others (e.g. Contract net). As opposed to that, in a cooperative architecture the different types of agents each specialize to solve a part of the problem, and work together to combine their capabilities (e.g. Specification sharing, blackboard system). [94] suggests an approach for a cooperative multi-agent architecture where agents are divided into two groups, task-specific and information-specific. Task-specific agents specialize in managing a certain task and usually interact with the user. Information specific agents specialize in managing access to an information source and offer this functionality to other agents. Advantages like higher flexibility or reusability are well known from classic ideas of distribution, modularity or object-oriented approaches. Many users can share the functionality of agents, new functionality can be added easily and agent modules can be re-used which allows an easier development of similar agents.

A MAS is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity. More recently, the term multi-agent system has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics[36]:

- An individual agent has incomplete capabilities to solve a problem
- There is no global system control
• Data is decentralized

• Computation is asynchronous

• Agents socialize with each other either to cooperate or to compete.

An information agent is an agent that has access to at least one and potentially many information sources, and is able to collate and manipulate information obtained from these sources in order to answer queries posed by users and other information agents.

A Cooperative Information System (CIS) is considered as a cooperative multi-agent system integrated by a set of agents, data, and procedures working, in a cooperative way, to support daily activities in the organisation. They have a common goal, exchange information, and work together in order to achieve their objective.

Agents can socialise using a rich set of standard interaction patterns. Communication enables the agents to coordinate their actions and behaviour, resulting in systems that are more coherent. Coordination involves cooperation, planning (centralised and distributed). Agent communication also involves knowledge exchange using a higher-level semantic model that is often based on ontologies. A multi-agent system is a good potential architecture for integrating heterogeneous databases in that agents are naturally distributed and autonomous; they can use rich explicit communication protocols to interoperate and they can naturally link to semantic models to help resolve interoperability problems. Multi-agent systems have been and are the subject of a very active research community.
1.2 Information Retrieval Systems

Information retrieval (IR) is the science of searching for information in documents, searching for documents, searching for metadata which describe documents, or searching within databases, whether relational stand-alone databases or hypertextually-networked databases such as the World Wide Web. Information Retrieval systems are designed with the objective of providing, in response to a user query, references to documents that would contain the information desired by the user.

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy. An object is an entity that is represented by information in a database. User queries are matched against the database information. Depending on the application the data objects may be, for example, text documents, images, audio, mind maps or videos. Often the documents themselves are not kept or stored directly in the IR system, but are instead represented in the system by document surrogates or metadata. Most IR systems compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value the top ranking objects are then shown to the user the process may then be iterated if the user wishes to refine the query.

For the information retrieval to be efficient, the documents are typically transformed into a suitable representation. An information retrieval model specifies representations used for documents and queries, and how they are compared (Turtle and Croft, 1992). Common representation models are categorized into two dimensions: the mathematical basis and the properties of the model (Kuropka, 2004), as shown in Figure 1.1.

For effectively retrieving relevant documents by IR strategies, the documents
are typically transformed into a suitable representation. Each retrieval strategy incorporates a specific model for its document representation purposes. The Figure 1.1 illustrates the relationship of some common models. In the picture, the models are categorized according to two dimensions: the mathematical basis and the properties of the model.

![Figure 1.1: Categorization of Information retrieval](image)

### 1.2.0.1 Mathematical basis

- Set-theoretic models represent documents as sets of words or phrases. Similarities are usually derived from set-theoretic operations on those sets. Common models are:
  - Standard Boolean model
  - Extended Boolean model
  - Fuzzy retrieval

- Algebraic models represent documents and queries usually as vectors, matrices, or tuples. The similarity of the query vector and document vector is represented as a scalar value.
- Vector space model
- Generalized vector space model
- (Enhanced) Topic-based Vector Space Model
- Extended Boolean model
- Latent semantic indexing aka latent semantic analysis

- Probabilistic models treat the process of document retrieval as a probabilistic inference. Similarities are computed as probabilities that a document is relevant for a given query. Probabilistic theorems like the Bayes' theorem are often used in these models.
  - Binary Independence Model
  - Probabilistic relevance model on which is based the okapi (BM25) relevance function
  - Uncertain inference
  - Language models
  - Divergence-from-randomness model
  - Latent Dirichlet allocation

- Feature-based retrieval models view documents as vectors of values of feature functions (or just features) and seek the best way to combine these features into a single relevance score, typically by learning to rank methods. Feature functions are arbitrary functions of document and query, and as such can easily incorporate almost any other retrieval model as just a yet another feature.

1.2.1 Properties of the model

- Models without term-interdependencies treat different terms/words as independent this fact is usually represented in vector space models by the orthogonality assumption of term vectors or in probabilistic models by an in dependency assumption for term variables.
• Models with immanent term interdependencies allow a representation of interdependencies between terms. However the degree of the interdependency between two terms is defined by the model itself. It is usually directly or indirectly derived (e.g. by dimensional reduction) from the co-occurrence of those terms in the whole set of documents.

• Models with transcendent term interdependencies allow a representation of interdependencies between terms, but they do not allege how the interdependency between two terms is defined they relay an external source for the degree of interdependency between two terms.

The purpose of this section is to cover three of the most important classic text IR models, namely: Boolean, Vector and Probabilistic. In the Boolean model document and queries are represented as a set of index terms. In the Vector-space model documents and queries are represented as vectors in a t-dimensional space. In the basic probabilistic model documents and queries representations are based on probability theory. Following the definition in (Baeza Yates and Ribeiro Neto, 1999) an IR model is a quadruple \( [D, Q, F, \text{sim}] \), where:

- \( D \) is a set of (logical representations of) documents
- \( Q \) is a set of (logical representations of) queries.
- \( F \) is a framework for modeling documents, queries, and their relationships.

where \( U \) is a totally ordered set this ranking and the total order in \( U \) define an order in the set of documents, for a fixed query.

### 1.2.2 Boolean model

The Boolean Model is a simple retrieval model based on set theory and Boolean algebra. Documents are represented by the index terms extracted from documents, and queries are Boolean expressions on terms. Following the previous notation, here:
• D the elements of D are represented as sets of index terms occurring in each document. Terms are treated as logic propositions, denoting whether the term is either present (1) or absent (0) in the document. Documents can thus be seen as the conjunction of their terms.

• Q queries are represented as a Boolean expression composed by index terms and logic operators (AND, OR, NOT) which can be normalized to a disjunction of conjunctive vectors (i.e., in DNF2, disjunctive normal form).

• F is a Boolean algebra over sets of terms and sets of documents.

• sim is defined by considering that a document is predicted to be relevant to a query if its index terms satisfy the query expression.

### 1.2.3 Vector-space model

The vector-space model (VSM) recognizes that the use of binary weights is too limiting and proposes a framework in which partial matching is possible. This is accomplished by assigning non-binary weights to index terms in queries and documents; these terms' weights are ultimately used to compute the degree of similarity between each document stored in the system and the user query. By sorting the retrieved documents in decreasing order of this degree of similarity, the VSM takes into consideration documents which match the query terms only partially. The main resulting effect is that the ranked document answer set is considerably more precise (in the sense that it better matches the user information need) than the answer set retrieved by a Boolean model. Following the previous notation:

• D documents are represented by a vector of words or index terms occurring in the document. Each term in the document, each pair \((t_i, d_j)\) has a positive, non-binary associated weight \(w_{i,j}\).

• Q queries are represented as a vector of words or index terms occurring in the query. Each term in the query, each pair \((t_i, q)\) has a positive, non-binary associated weight \(w_{i,q}\).
• F is an algebraic model over vectors in a t-dimensional space.

• sim estimates the degree of similarity of a document d to a query q as the correlation between the vectors djj and q this correlation can be quantified, for instance, by the cosine of the angle between the two vectors:

Using the cosine the similarity between document dj and query q can be calculated as:

\[ \text{sim}(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j}w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \]  \hspace{1cm} (1.1)

1.2.4 Probabilistic model

The probabilistic model aims to capture the IR problem in a probabilistic framework the fundamental idea is as follows. Given a query q and a collection of documents D, a subset R of D is assumed to exist which contains exactly the relevant documents to q (the ideal answer set). The probabilistic retrieval model then ranks documents in decreasing order of probability of belonging to this set (i.e. of being relevant to the information need), which is noted as \( P(R - q, d_j) \), where d is a document in D. Following the previous notation:

• D documents are represented as a vector of words or index terms occurring in a document. Each term in the document, that is, each pair \( (t_i, d_j) \), has a binary associated weight 1 or 0, denoting the presence or absence of the term in the document.

• Q queries are represented by a vector of words or index terms that occur in the query. Each term in the query, that is, each pair \( (t_i, q) \) has a binary weight 1 or 0, denoting the presence or absence of the term in the query.

• F is a probabilistic model that ranks documents in order of probability of relevance to the query.

• sim measures the degree of similarity of a document dj to a query qij as the probability of d to be part of the subset R of relevant documents for
q. this is measured in the probabilistic model as the odds of relevance, as given by

\[ \text{sim}(d_j, q) = \frac{P(R|d_j)}{P(\bar{R}|d_j)} \]  \hspace{1cm} (1.2)

1.2.5 Other models

Over the years, alternative modelling paradigms have been proposed. Among those models we can highlight: the fuzzy and the extended Boolean models, the generalized vector model, the neural network models, etc. An introduction to these models can be found in (Baeza Yates and Ribeiro Neto, 1999). More recently, so-called Language Models have become popular and widely studied in the IR research field, because of their good performance and the fact that they unify term weighting and result ranking in a single model with probabilistic foundation. (Ponte & Croft, 1998).

1.2.6 Performance and correctness measures

Many different measures for evaluating the performance of information retrieval systems have been proposed; the measures require a collection of documents and a query. All common measures described here assume a ground truth notion of relevancy: every document is known to be either relevant or non-relevant to a particular query. In practice, queries may be ill-posed and there may be different shades of relevancy.

1.2.6.1 Precision

Precision is the fraction of the documents retrieved that are relevant to the user’s information need.

\[ \text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \]  \hspace{1cm} (1.3)

In binary classification, precision is analogous to positive predictive value. Precision takes all retrieved documents into account. It can also be evaluated at a given cut-off rank, considering only the topmost results.
returned by the system this measure is called precision at n or P. Note that the meaning and usage of "precision" in the field of Information Retrieval differs from the definition of accuracy and precision within other branches of science and statistics.

1.2.6.2 Recall

Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

\[
\text{Recall} = \frac{|\text{relevant documents} \cap \text{retrieved documents}|}{|\text{relevant documents}|} \quad (1.4)
\]

In binary classification, recall is often called sensitivity. So it can be looked at as the probability that a relevant document is retrieved by the query. It is trivial to achieve recall of 100% by returning all documents in response to any query; therefore recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by computing the precision.

1.2.6.3 F-measure

The weighted harmonic mean of precision and recall, the traditional F-measure or balanced F-score is:

\[
F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (1.5)
\]

This is also known as the $F_1$ measure, because recall and precision are evenly weighted. The general formula for non-negative real $\beta$ is:

\[
F_\beta = \frac{(1 + \beta^2) \cdot (\text{Precision} \cdot \text{Recall})}{(\beta^2 \cdot \text{Precision} + \text{Recall})} \quad (1.6)
\]

Two other commonly used F measures are the $F_2$ measure, which weights recall twice as much as precision, and the $F_{0.5}$ measure, which weights precision twice as much as recall. The F-measure was derived by van Rijsbergen (1979)
so that $F_\beta$ measures the effectiveness of retrieval with respect to a user who attaches $\beta$ times as much importance to recall as precision. It is based on van Rijsbergen’s effectiveness measure.

$$E = 1 - \frac{1}{\alpha P + \frac{1-\alpha}{R}}$$

(1.7)

Their relationship is

$$F_\beta = 1 - E \text{ where } \alpha = \frac{1}{1 + \beta^2}$$

(1.8)

### 1.3 Semantic Web

The Semantic Web is a Web of actionable information - information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of meaning in which the logical connection of terms establishes interoperability between systems [82]. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The aims of Semantic Web are to structure the information in all kinds of data resource and applications and to promote more automatic machine-readable data and processing and hence improve IR efficiency. "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [8]. XML-based Ontology languages have been also proposed as Web based knowledge description languages [32].

Figure 1.2 taken from [82] shows the proposed layers of the Semantic Web, with the higher level languages using the syntax (and semantics) of the lower level languages. This thesis focuses primarily on the Ontology language level, and the sort of agent-based computing that they enable. Higher levels (with complex logics and the exchange of proofs to establish trust relationships) will enable even more interesting functionality.
Some of these levels in more detail are:

- Extensible Markup Language (XML) provides the syntax for structured documents, but imposes no semantic constraints on the meaning of these documents. XML Schema is a language for restricting the structure of XML documents.

- Resource Description Framework (RDF) is a metadata model for defining data structures called resources and relations between them and provides a simple semantics for a data model whose syntax is XML.
- RDF Schema or RDFS is a vocabulary for describing properties and classes of RDF resources that supports a more expressive semantics for generalization hierarchies of such properties and classes.

- DAML+OIL (DARPA Agent Mark-up Language + Ontology Inference Layer) is another extension to XML and to RDF (DAML+OIL) that provides a richer set of constructs to create Ontology conceptual data models and to mark-up information so that it is machine readable and understandable. A subset of First Order Logic has been merged into the Ontology model to support logic processing and operating is allowed.

- Web Ontology Language (OWL) has been defined as a replacement to supersede DAML+OIL and adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. Disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. Symmetry), and enumerated classes.

### 1.3.1 RDF and RDFS

Statements are the main metadata concept in the RDF model and can be used to link two resources together. Hence statements specify triples of a verb (or predicate or property) that links a subject resource to an object or value the verb may also be specified as a resource. Hence triple statements specify subject-verb-object or subject-predicate-value relationships. Each RDF statement can be stored as a relational database table whose name is the predicate and whose subject-value instances form throws in the table.

RDFS (RDF Schema) is a language for describing ontologies. RDFS defines basic classes for resources, properties, literals, containers, container member properties and classes of properties such as sub-classes, domains, ranges and labels. RDFS supports many of the above properties and can be considered an Ontology language.
1.3.2 Ontology

Ontology’s are conceptual models that can be used for knowledge sharing. An Ontology is characterized by the explicitness of the conceptual model and richness of the structures used, to represent and manage knowledge, information and services the model and the structures will also influence the degree of flexibility of the computation or inference that applications can derive from it.[87]defines an Ontology in the following way: the subject of Ontology is the study of the categories of things that exist or may exist in some domain the product of such a study, called Ontology, is a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D the combination of logic with Ontology provides a language that can express relationships about the entities in the domain of interest.

Unlike data models, ontology’s are usually formed to be relatively independent and reusable across particular applications, i.e.the Ontology consists of generic knowledge that can be used by the different kinds of applications and tasks. There are many proposed Ontology models. Regardless of the properties of the specific Ontology, ontologies in general include the following elements:

- Taxonomic relations between classes.
- Data type properties, descriptions of attributes of elements of classes.
- Object properties, descriptions of relations between elements of classes.
- Instances of classes and properties.

Data type properties and object properties are collectively referred to as the properties of a class. A set of assertions about the loaded into a reasoning system is called a knowledge base (KB) these assertions may include facts about individuals that are members of classes, as well as various derived facts, facts not literally present in the original textual representation
of the Ontology, but entailed (logically implied) by the semantics of the particular Ontology language.

These assertions may be based on a single Ontology or multiple distributed ontologies that have been combined using defined mechanisms. Semantics is the set of formalized concept and relations that have been defined to describe the logic representation with the given restriction so that the logic application can read, understand, process and deduce the logic relations from the defined knowledge base in order to answer information queries in a more intelligent way. Most of applications are designed to handle the case in the particular domain and application. The logic inference, reuse and reasoning in such application are quite limited[80].

There are many Ontology representations that can be chosen. Ontologies started to gain widespread interest and support as part of an initiative called the Semantic Web. The Semantic Web covers a range of XML-based approaches such as RDFS, as it supports the above Ontology features, DAML+OIL and OWL. At the start of the PhD in late 2002, DAML+OIL was the most widely used and supported Ontology Model.

1.4 Web Mining

This Section gives an explanation of what web usage mining is, its roots, methods and an overview of the web usage mining process

1.4.1 Data Mining

Web usage mining is a subset of web mining operations which itself is a subset of data mining in general the aim is to use the data and information extracted in web systems in order to reach knowledge of the system itself. To
better understand the concepts brief definitions of keywords can be given as [104]: Data: A class of information objects, made up of units of binary code that are intended to be stored, processed, and transmitted by digital computers. Information: is a set of facts with processing capability added, such as context, relationships to other facts about the same or related objects, implying an increased usefulness. Information provides meaning to data. Knowledge: is the summation of information into independent concepts and rules that can explain relationships or predict outcomes.

Data mining is a set operations performed on a collection of data or a subset of it so as to extract meaningful patterns on the data. Another definition is Data mining is the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data [96]. That is, data mining attempts to extract knowledge from data. If a subset is to be used, careful and unbiased sampling algorithms should be used to avoid biased result. Data mining is different from information extraction although they are closely related. Information extraction is the process of extraction information from data sources whether they are structured, unstructured or semi-structured into structured and computer understandable data formats. Data mining operations are performed on the data already extracted by means of information retrieval.

Based on the types of data sources it is applied on, data mining can be categorized. One such application would be on geographic data such as digital maps as usually seen most GIS applications which is called spatial data mining. Another area where data mining is widely used is bioinformatics where very large data about protein structures, networks and genetic material is analysed. The sub category of interest in this thesis is the web mining which acts on the data made available in the World Wide Web (WWW) data servers.
1.4.2 Web Mining

Web mining consists of a set operations defined on data residing on WWW data servers. Mobasher et al. [14] defines web mining as the discovery and analysis of useful information from the World Wide Web. Such data can be the content presented to users of the Websites such as hyper text markup language (HTML) files, images, text, audio or video. Also the physical structure of the Websites or the server logs that keep track of user accesses to the resources mentioned above can be targets of web mining techniques. Web mining as a sub category of data mining is fairly recent compared to other areas since the introduction of Internet and its widespread usage itself is also recent. However, the incentive to mine the data available on the Internet is quite strong. Both the number of users around the world accessing online data and the volume of the data itself motivate the stakeholders of the Websites to consider analysing the data and user behaviour.

Web mining is mainly categorized into two subsets namely web content mining and web usage mining [14]. While the content mining approaches focus on the content of single Webpages, web usage mining uses server logs that detail the past accesses to the Website data made available to public. Usually the physical structure of the Website itself which is a graph representation of all Webpages in the Website is used as a part of either method. However recent approaches [17] that appoint more focus on the physical link structure of the Website have introduced web structure mining as a separate concept. In order to understand the differences a brief description and area of work for each category is summarized below.

1.4.3 Web Content Mining

Web content mining describes the automatic search of information resources available on-line. [58] The focus is on the content of Webpages themselves. Mobasher [14] categorizes content mining as agent-based approaches; where intelligent web agents such as crawlers autonomously crawl
the web and classify data [89] and database approaches; where information retrieval tasks are employed to store web data in databases where data mining process can take place [45]. Most web content mining studies have focused on textual and graphical data since the early years of internet mostly featured textual or graphical information. Recent studies started to focus on visual and aural data such as sound and video content too.

1.4.4 Web Structure Mining

One of the most well known algorithms, Page Rank Measure [68] and Hubs and Authorities [41] are based on the links between pages. Web structure mining focuses on the links rather than the content of the pages, their usage or semantics. [91] Divides links into two categories the hyperlinks that link the Webpages and the document structure itself such as the xml or html structure. [58] describe details the latter.

1.4.5 Web Usage Mining

The main topic of this thesis is the web usage mining. Usage mining as the name implies focus on how the users of websites interact with Website, the Webpages visited, the order of visit, timestamps of visits and durations of them.

The main source of data for the web usage mining is the server logs which log each visit to each Webpage with possibly IP, referrer, time, browser and accessed page link. Although many areas and applications can be cited where usage mining is useful, it can be said the main idea behind web usage mining is to let users of a Website to use it with ease efficiently, predict and recommend parts of the Website to user based on their and previous users actions on the Website.
1.4.6 Web Mining for Personalization

Web Personalization is the process of customizing a Website to the needs of specific users, taking advantage of the knowledge acquired from the analysis of the users navigational behavior (usage data) in correlation with other information collected in the Web context, namely structure, content and user profile data. Due to the explosive growth of the Web, the domain of Web Personalization has gained great momentum both in the research and the commercial area. In this chapter present a survey of the use of Web mining for Web Personalization. More specifically, introduce the modules that comprise a Web Personalization system, emphasizing on the Web usage mining module. A review of the most common methods that are used as well as technical issues that occur is given, along with a brief overview of the most popular tools and applications available from software vendors. Moreover, the most important research initiatives in the Web usage mining and personalization area are presented.

The continuous growth in the size and use of the World Wide Web imposes new methods of design and development of on-line Information Services. Most Web structures are large and complicated and users often miss the goal of their inquiry, or receive ambiguous results when they try to navigate through them. On the other hand, the e-business sector is rapidly evolving and the need for Web market places that anticipate the needs of the customers is more than ever evident therefore, the requirement for predicting user needs in order to improve the usability and user retention of a Website can be addressed by personalizing it. Web Personalization is defined as any action that adapts the information or services provided by a Website to the needs of a particular user or a set of users, taking advantage of the knowledge gained from the users navigational behaviour and individual interests, in combination with the content and the structure of the Website. The objective of a Web Personalization system is to provide users with the information they want or need, without expecting from them to ask for it explicitly.
Principal elements of Web Personalization include (a) the categorization and pre-processing of Web data, (b) the extraction of correlations between and across different kinds of such data and (c) the determination of the actions that should be recommended by such a personalization system.

Web data are those that can be collected and used in the context of Web Personalization. These data are classified in four categories according to:

- **Content data** are presented to the end-user appropriately structured. They can be simple text, images or structured data, such as information retrieved from databases.

- **Structure data** represent the way content is organized. They can be either data entities used within a Webpage, such as HTML or XML tags, or data entities used to put a Website together, such as hyperlinks connecting one page to another.

- **Usage data** represent a Website’s usage, such as visitors IP address, time and date of access, complete path (files or directories) accessed, referrers address, and other attributes that can be included in a Web access log.

- **User profile data** provide information about the users of a Website. A user profile contains demographic information (such as name, age, country, marital status, education, interests etc) for each user of a Website, as well as information about users interests and preferences. Such information is acquired through registration forms or questionnaires, or can be inferred by analyzing Web usage logs.

The overall process of usage-based Web Personalization consists of five modules, which correspond to each step of the process these are:

- **User profiling:** In the Web domain, user profiling is the process of gathering information specific to each visitor, either explicitly or implicitly. A user profile includes demographic information about the user, their interests and even their behaviour when browsing a
Website. this information is exploited in order to customize the content and structure of a Website to the visitors specific and individual needs.

- Log analysis and Web usage mining: This is the procedure where the information stored in Web server logs is processed by applying data mining techniques in order to (a) extract statistical information and discover interesting usage patterns, (b) cluster the users into groups according to their navigational behaviour, and (c) discover potential correlations between Webpages and user groups. This process of extracting information concerning the browsing behaviour of the users can be regarded as part of the user profiling process. It is therefore evident that the user profiling and Web usage mining modules overlap.

- Content management: This is the process of classifying the content of a Website in semantic categories in order to make information retrieval and presentation easier for the users. Content management is very important for Websites whose content is increasing on a daily basis, such as news sites or portals.

- Website publishing: A publishing mechanism is used in order to present the content stored locally in a Web server and/or some information retrieved from other Web resources in a uniform way to the end-user. Different technologies can be used to publish data on the Web.

- Information acquisition and searching: In many cases information provided by a Website, is not physically stored in the Websites server. In the case of a Web portal (vertical portal), users are interested in information from various Web sources. It remains to the Website editors to search the Web for content of interest that should consequently be classified into thematic categories. Searching and relevance ranking techniques must be employed both in the process of acquisition of relevant information and in the publishing of the appropriate data to each group of users.
1.5 Web Personalization

Website personalization can be defined as the process of customizing the content and structure of a Website to the specific and individual needs of each user taking advantage of the users navigational behaviour the steps of a Web Personalization process include: (a) the collection of Web data, (b) the modelling and categorization of these data (pre-processing phase), (c) the analysis of the Collected data and (d) the determination of the actions that should be performed the ways that are employed in order to analyse the collected data include content-based filtering, collaborative filtering, rule-based filtering and Web usage mining. The site is personalized through the highlighting of existing hyperlinks, the dynamic insertion of new hyperlinks that seem to be of interest for the current user, or even the creation of new index pages.

Content-based filtering systems are solely based on individual users preferences. the system tracks each users behaviour and recommends them items that are similar to items the user liked in the past. Understanding the meaning of Web content and, more importantly, how it relates to the real meaning of the users query, is a crucial step in the retrieval process. Figure shows the principal content-based personalization approaches.

Collaborative filtering systems invite users to rate objects or divulge their preferences and interests and then return information that is predicted to be of interest for them. This is based on the assumption that users with similar behaviour (for example users that rate similar objects) have analogous interests.

In rule-based filtering the users are asked to answer to a set of questions these questions are derived from a decision tree, so as the user proceeds on answering them, what she/he finally receives as a result (for example a list of products) is tailored to their needs.
Content-based, rule-based and collaborative filtering may also be used in combination, for deducing more accurate conclusions.

1.5.1 Web usage mining

This process relies on the application of statistical and data mining methods to the Web log data, resulting in a set of useful patterns that indicate users navigational behaviour. The data mining methods that are employed are: association rule mining, sequential pattern discovery, clustering and classification. This knowledge is then used from the system in order to personalize the site according to each user's behaviour and profile.

The block diagram illustrated in Figure 1.3 represents the functional architecture of a Web Personalization system in terms of the modules and data sources that were described earlier. The content management module processes the Websites content and classifies it in conceptual categories. The Websites content can be enhanced with additional information acquired from other Web sources, using advanced search techniques. Given the site map structure and the usage logs, a Web usage miner provides results regarding usage patterns, user behaviour, session and user clusters, click-stream information etc. Additional information about the individual users can be obtained by the user profiles. Moreover, any information extracted from the Web usage mining process concerning each user's navigational behaviour can then be added to his/her profile. All this information about nodes, links, Web content, typical behaviours and patterns is conceptually abstracted and classified into semantic categories. Any information extracted from the interrelation between knowledge acquired using usage mining techniques and knowledge acquired from content management will then provide the framework for evaluating possible alternatives for restructuring the site. A publishing mechanism will perform the site modification, ensuring that each user navigates through the optimal site structure. The available content options for each user will be ranked according to user's interests.
1.5.2 User Modeling in Personalized Systems

Tracking what pages the user has chosen to visit and their submitted queries is a type of user modeling or profiling technique, from which important features of users are learned and then used to get more relevant information. Most of the personalized search systems employ a user modelling component that occurs during the information retrieval or filtering. Basically, this is the major component needed to provide tailored results that satisfy the particular needs of single users.

In the simplest cases, user models consist of a registration form or a questionnaire, with an explicit declaration of interest by the user. In more complex and extended cases, A.Micarelli et al. a user model consists of dynamic information structures that take into account background information, such as educational level and the familiarity with the area of interest, or how the user behaves over time. For example, the ifWeb
prototype[5] makes use of user models based on semantic networks in order to create a representation of the available topics of interests. It supports users during Web surfing, acting as hypermedia search assistant.

As an example of a very simple personalized search tool, GOOGL Es Alerts is an agent that automatically sends emails to the user each time new results for given query terms become available, both from the Web and News sites. GOOGL Es Alerts builds user models using an explicit approach where users explicitly construct the model by describing the information in which they are interested in. In this particular case, the user suggests a set of keywords, sometimes called routing query, which must appear in the retrieved documents, thus filtering the information stream. As soon as new information is published on the Web, the system evaluates it according to the stored profile, alerting the user of such new and potentially interesting contents. The obtained profiles are relatively simple and act as standard queries. Since the routing queries are suggested by users and the results are never adapted by the system to particular needs or tasks, the system’s personalization is really limited.

A further tool named GOOGL Es Personalized Search used to deliver customized search results based on user profiles overcomes some of the Alerts problems. The results were instantly rearranged by dragging a series of sliders that define the personalization level concerning pre-defined sets of topics. Basically, while indexing, the engine categorizes pages collected from the Web according to a topic taxonomy. When users submit a query, the system looks through pages associated with their interests, that is, the selected topics, to find matches affecting the search results. Due to the kind of feedback employed to build the profiles, the user is still required to point the system to the information that is considered most interesting or, in some cases, suggesting data to be ignored in the future. For this reason this tool has been replaced with a new technology.
In personalized search systems the user modelling component can affect the search in three distinct phases, showed in Figure 1.4

- **Part of retrieval process:** the ranking is a unified process wherein user profiles are employed to score Web contents.

- **Re-ranking:** user profiles take part in a second step, after evaluating the corpus ranked via non-personalized scores.

- **Query modification:** user profiles affect the submitted representation of the information needs, e.g., query, modifying or augmenting it.

The first technique is more likely to provide quick query response, because the traditional ranking system can be directly adapted to include personalization, avoiding repeated or superfluous computation. However, since the personalization process usually takes a long time compared with traditional non-personalized IR techniques, most search engines do not employ any personalization at all time constraints that force the system to provide result lists in less than a second cannot be met for all users. On the
other hand, re-ranking documents as suggested by an external system, such as a search engine, allows the user to selectively employ personalization approaches.

A user profile can be either static, when the information it contains is never or rarely altered (e.g., demographic information), or dynamic when the user profile’s data change frequently. Such information is obtained either explicitly, using on-line registration forms and questionnaires resulting in static user profiles, or implicitly, by recording the navigational behaviour and/or the preferences of each user, resulting in dynamic user profiles (Figure 1.5). In the latter case, there are two further options: either regarding each user as a member of a group and creating aggregate user profiles, or addressing any changes to each user individually. When addressing the users as a group, the method used is the creation of aggregate user profiles based on rules and patterns extracted by applying Web usage mining techniques to Web server logs. Using this knowledge, the Website can be appropriately customized.

Figure 1.5: Implicit and Explicit Feedback are used to learn and keep updated the profile of the user used during the personalization
1.5.3 Log Analysis and Web Usage Mining

The purpose of Web usage mining is to reveal the knowledge hidden in the log files of a Web server. By applying statistical and data mining methods to the Web log data, interesting patterns concerning the users navigational behaviour can be identified, such as user and page clusters, as well as possible correlations between Webpages and user groups. The Web usage mining process can be regarded as a three-phase process, consisting of the data preparation, the pattern discovery and the pattern analysis phases. In the first phase, Web log data are pre-processed in order to identify users, sessions, page views etc. In the second phase statistical methods, as well as data mining methods (such as association rules, sequential pattern discovery, clustering and classification) are applied in order to detect interesting patterns. These patterns are stored so that they can be further analysed in the third phase of the Web usage mining process.

A description of the fields included in a log entry of a Web usage log follows, along with a set of definitions of Web data abstractions, such as Website, user, session, page views and click-streams.

1.5.4 Web usage mining

Log analysis is regarded as the simplest method used in the Web usage mining process the purpose of Web usage mining is to apply statistical and data mining techniques to the pre-processed Web log data, in order to discover useful patterns. The most common and simple method that can be applied to such data is statistical analysis. More advanced data mining methods and algorithms tailored appropriately for being used in the Web domain include association rules, sequential pattern discovery, clustering and classification.

Association rule mining is a technique for finding frequent patterns, associations and correlations among sets of items. Association rules are used in order to reveal correlations between pages accessed together during a
server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. Aside from being exploited for business applications, such observations also can be used as a guide for Website restructuring, for example by adding links that interconnect pages often viewed together, or as a way to improve the systems performance through pre-fetching Web data.

Sequential pattern discovery is an extension of association rules mining in that it reveals patterns of co-occurrence incorporating the notion of time sequence. In the Web domain such a pattern might be a Webpage or a set of pages accessed immediately after another set of pages. Using this approach, useful users trends can be discovered, and predictions concerning visit patterns can be made.

Clustering is used to group together items that have similar characteristics. In the context of Web mining, we can distinguish two cases, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users perception. User clustering results in groups of users that seem to behave similarly when navigating through a Website. Such knowledge is used in e-commerce in order to perform market segmentation but is also helpful when the objective is to personalize a Website.

Classification is a process that maps a data item into one of several predetermined classes. In the Web domain classes usually represent different user profiles and classification is performed using selected features that describe each users category the most common classification algorithms are decision trees, naive Bayesian classifier, neural networks etc.

After discovering patterns from usage data, a further analysis has to be conducted the exact methodology that should be followed depends on the technique previously used the most common ways of analyzing such patterns
are either by using a query mechanism on a database where the results are stored, or by loading the results into a data cube and then performing OLAP operations.

Additionally, visualization techniques are used for an easier interpretation of the results. Using these results in association with content and structure information concerning the Website there can be extracted useful knowledge for modifying the site according to the correlation between user and content groups.