Ontological technologies for user modelling

Sergey Sosnovsky*

School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260, USA
E-mail: sosnovsky@gmail.com
*Corresponding author

Darina Dicheva

Department of Computer Science, Winston-Salem State University, Winston Salem, NC 27110, USA
E-mail: dichevad@wssu.edu

Abstract: This paper brings together research from two different fields – user modelling and web ontologies – in attempt to demonstrate how recent semantic trends in web development can be combined with the modern technologies of user modelling. Over the last several years, a number of user-adaptive systems have been exploiting ontologies for the purposes of semantics representation, automatic knowledge acquisition, domain and user model visualisation and creation of interoperable and reusable architectural solutions. Before discussing these projects, we first overview the underlying user modelling and ontological technologies. As an example of the project employing ontology-based user modelling, we present an experiment design for translation of overlay student models for relative domains by means of ontology mapping.

Keywords: user modelling; ontologies; adaptation; SW; semantic web.


Biographical notes: Sergey Sosnovsky is a PhD candidate at the School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA. He received his MSc and BSc from Kazan State Technological University, Kazan, Russia. His research focuses on combining new trends of Web development with Adaptive and User Modelling technologies. He has co-authored about 60 peer-reviewed research publications and served on programming committees of several workshops in the area of Semantic Web for Adaptation.

Darina Dicheva is Paul Fulton Distinguished Professor of Computer Science at Winston-Salem State University, NC, USA. Her research interests include semantic web, social networks, information retrieval and filtering, user modelling, intelligent learning environments, and digital libraries. She has authored over 120 research papers and has been on the programme committees of more than 40 international conferences in the recent years. She is a co-organiser of the International Workshop on Semantic Web for e-Learning (SWEL) series.

1 Introduction

Among the promising directions of the World Wide Web (WWW) development, two paradigms can be named that share the ultimate goal of providing users with more efficient access to online information and services: the Adaptive Web (AW) and the Semantic Web (SW). The first aims at enabling different kinds of personalised information experiences from adaptive e-shopping to online intelligent tutoring. The second proposes a set of standards and technologies for meaningful (semantically-enriched) description and automatic discovery of data and knowledge on the web. Bringing these two fields together can result in a successful, mutually beneficial synergy. The AW challenges introduce interesting application areas for the SW technologies, while the SW vision, in its turn, opens new perspectives for the AW research.

In this review we do not attempt to describe all aspects of the AW-SW technology fusion, instead, we focus on analysing the novel projects emerged at the border of two core subfields of AW and SW: user modelling and web ontologies.

User modelling is much an older area of research than AW. The first systems modelling users were developed as long ago as 1970s. Nowadays, user modelling constitutes one of the major and perhaps the most challenging...
part of the development of AW applications. The precision of the modelling assumptions about the users defines the effectiveness of adaptive systems in general. An incorrect interpretation of a user leads to wrong adaptive decisions, which may result in user’s frustration, loss of trust, decrease in motivation to use the system, etc. Adequate representation of knowledge about the users, effective elicitation of user-related information, and utilisation of this information for organising coherent and meaningful adaptation are crucial factors for the success of AW systems.

The notion of web ontology is central for the SW initiative. Automatic discovery of information on the web and its machine-based interpretation are not possible unless the information is semantically-enriched with metadata providing the shared meaning of the content. Ontologies are the instrument to present and convey such meaning. Ontologies have been studied for many years, first by philosophers and logicians, and later by researchers in the field of Artificial Intelligence (AI) and Knowledge Representation (KR). SW, however, gave a new spin to this study and nowadays the field of web ontologies is attracting much of interest. The SW activity of the World Wide Web Consortium (W3C) leads and supports these efforts by developing a set of standards for ontology representation and processing on the web (W3C, 2001b).

Recently a lot of research has been done on the crossroad of user modelling and web ontologies. Ultimately, both disciplines attempt to model real world phenomena qualitatively: ontologies – a particular area of knowledge, and user modelling – the internal state of a human user. Many user-modelling approaches exploit content-based characteristics of users (users’ knowledge, interests, etc.) and hence can directly benefit of the high-quality domain models provided by ontologies. Besides, as the majority of user modelling projects have been deployed on the web and web ontologies are becoming de facto standard for WWW-based KR, the cooperation between user modelling and web ontologies seems inevitable.

The paper is organised as follows. The next section gives an overview of user modelling. It starts with a description of different user’s characteristics modelled by adaptive systems, followed by an analysis of the most important technologies for user model representation and elicitation. The set of technologies is chosen to demonstrate the applicability of ontologies to user modelling.

Section 3 is devoted to ontologies. It first discusses the origins of the ontological research and the requirements for web ontologies and then summarises central aspects of ontologies, including ontology representation, ontology-based knowledge acquisition, ontology-based annotation, ontology mapping, etc.

The fourth and central section examines a set of projects that merge user modelling and ontologies. It discusses such features of ontologies as controlled vocabulary of terms, available formats for sharable knowledge representation, well-defined structure of domain knowledge, support of logical reasoning, etc., that have been appreciated by the UM researchers and utilised for developing AW systems. The section attempts to draw the connection between certain technologies, for example, open user modelling and ontology visualisation, overlay user modelling and ontology representation, unobtrusive user modelling and ontology learning, etc. As this area of research is rather new, the section may look unbalanced. Some of its subsections are more detailed than others; for some problems solutions have not been proposed yet. More research efforts are needed for ontology-based user modelling technologies to mature.

Section 5 concludes the paper by mentioning not covered issues and technologies, as well as possible future trends.

2 User modelling

The field of user modelling came a long way from lab projects (Carbonell, 1970) to the development of dedicated commercial user modelling servers that support millions of users (see Fink and Kobsa, 2000) for a comprehensive review). This section provides a brief overview of the evolution of efforts in this field and lists the most important user modelling technologies developed over the years. It focuses on the user modelling trends that can benefit of integration with the field of web ontologies.

2.1 User modelling dimensions

The core idea of adaptation is based on the assumption that differences in some user characteristics affect the usefulness of the services or information provided to the individuals. Thus if a system’s behaviour is tailored according to such characteristics, its value to individuals will be increased. This section describes the most important characteristics with regard to adaptation.

2.1.1 Knowledge, beliefs, and background

These characteristics are especially important for the adaptive systems modelling students. Among all kinds of user-adaptive systems, Adaptive Educational Systems (AES) have the longest history of research. This class of adaptive systems is the most diverse and numerous. To name a few, it includes ICAI systems (e.g., Scholar (Carbonell, 1970), Cognitive Tutors (e.g., Lisp-Tutor (Corbett and Anderson, 1994)), and Adaptive Educational Hypermedia Systems (e.g., Interbook (Brusilovsky et al., 1996)). For any of these systems student’s knowledge is the main characteristic defining system’s adaptivity. The most popular approach to modelling student knowledge relies on a fine-grained conceptual structure of a learning domain; thus, the aggregate student model consists of indicators for the student’s knowledge of particular domain concepts.

Different systems manage differently the evidence of incorrect knowledge or misconceptions. Some AEs simply accumulate this negative evidence in the knowledge assessment for corresponding concepts (Galeev et al., 1994)
or simply mark concepts as having a misconception (Mabbott and Bull, 2004), other systems support a special ‘buggy model’ (Vassileva, 1997). Some authors argue that the student model of an AES should not directly differentiate between correct and incorrect knowledge of a concept and treat them instead as student beliefs (Self, 1988).

Another relevant characteristic of a student is her or his background. Traditionally, background is defined as relevant experience gained outside the system, prior to using it. Unlike knowledge models, a background model is usually static (it does not change over time) and coarse-grained. Most typically, it is represented as a single parameter with several possible values (e.g., ‘novice’, ‘advanced’, ‘expert’) (Horvitz et al., 1998) or as a set of stereotypes (Rich, 1979).

### 2.1.2 Interests and preferences

These two characteristics have been often used as synonyms, especially in the context of AW systems. Several types of adaptive systems have been developed to assist information harvesting on the web, including adaptive recommenders (Pazzani and Billsus, 2007), adaptive search engines (Micarelli et al., 2007), and adaptive browser agents (Lieberman, 1995). The most important user characteristics for such systems are user’s information preferences or information interests. When modelling users’ preferences/interests the systems typically distinguish between long-term and short-term models. Long-term interests are relatively stable; they evolve slowly along the entire period of user’s working with the system, or are provided by the user explicitly in the form of general categories. A short-term model of interests is dynamic and usually populated and used during a single session, reflecting user’s current information task. The short-term model is discarded when the session ends.

#### 2.1.3 Goals, plans, tasks and needs

Modelling users’ goals and plans has been widely exploited in intelligent dialog systems. Knowledge of what situation a user is trying to achieve (goal) and what sequence of actions she or he is going to take on the way to the desired state of affairs (plan) is essential for such systems to maintain adaptive conversation with the user (Kass and Finin, 1988). Very close to these modelling dimensions is the concept of task or need. Modelling users’ information needs or information seeking tasks is a very popular approach employed by different adaptive systems on the web, such as adaptive recommender systems (McNee et al., 2006). AES model a student’s goal/task from a different perspective. Since the general student’s goal is clear – to learn the material – they do not try to recognise it but aim at finding the best strategy leading to the most efficient learning (see, for example, Conati et al. (1997)).

#### 2.1.4 Demographic information

For some kinds of adaptive systems, it is vital to be aware of demographic characteristics of a user from the very basic features, like gender, age or native language to more complex socio-cultural parameters, such as level of formal education and family income. Adaptation to demographic information is widely used in adaptive e-commerce systems (Bowne, 2000) and personalised ubiquitous applications (Fink and Kobsa, 2002). It can be also important in educational setting (Desimone, 1999).

### 2.1.5 Emotional state

Recently, a new class of technologies combined under the name ‘affective computing’ has gained a lot of interest in the field of user modelling (Picard and Klein, 2002). An adaptive system capable of modelling a certain user’s emotions obtains one more source for proper tailoring and justification of its behaviour. For example, the authors of Rodrigo et al. (2007) have found that such emotions as boredom and confusion lead students to an off-task behaviour, namely, gaming the adaptive system. One of the main challenges in affective computing is how to recognise different kinds of user’s emotions (Picard, 2003).

### 2.1.6 Context

Users’ context modelling is becoming an important direction of research as the interest in the development of adaptive applications for pervasive and ubiquitous settings is growing. The context of a user is a very broad concept; it can include any information about a user’s location, time, physical and social environment, the device being used to access the system, etc. Currently, the most popular class of applications adapting to the user’s context is various kinds of personalised guides and tours. For example, the GUIDE system (Cheverst et al., 2000) creates an adaptive tour of the city of Lancaster by utilising the current location of the user (to navigate to the most appropriate city attraction) and the time of the day (to match the hours when the attractions are open for visitors).

### 2.2 User model representation

This section discusses how the selected characteristics of a user can be modelled. Two classic approaches – overlay user modelling and user modelling via stereotypes – are presented. In addition, we describe the keyword-based representation of user models, which became popular as the web-based adaptive information retrieval technologies matured. Finally, we briefly summarise other important technologies for user model representations, which are less relevant to the main topic of this survey.

#### 2.2.1 Overlay user modelling

This is the oldest approach to the user model representation. Traditionally, it was employed by different kinds of AES for modelling student knowledge as a subset of domain expert knowledge. In their classic paper, Carr and Goldstein coined the term and provided the first definition of overlay user modelling, as a technique for “describing a student’s skills”
as “a set of hypotheses regarding the student’s familiarity” with the elements of domain representation provided by an expert (Carr and Goldstein, 1977). However, the main idea of gaining the domain of discourse into elementary components and using them for evaluation of students’ knowledge was first proposed in the Scholar system (Carbonell, 1970). These components have been named differently by different authors: topics, knowledge elements, learning outcomes, and – the most widely used – concepts.

In this context, a concept represents an atomic piece of declarative domain knowledge, coherent and semantically complete. An aggregate of concepts form the domain model. The overlay user model relies on the domain model as on a template and, essentially, consists of a set of concept-value pairs, where the value represents an assessment of the template and, essentially, consists of a set of concept-value pairs, where the value represents an assessment of the particular concept. Some systems also apply the overlay approach to representing user preferences or interests (e.g., De Bra et al., 2003).

The benefit of the overlay user model is its precision and flexibility. Fine-grained concept-based modelling allows systems to adjust their actions on a very detailed level. An overlay model is capable to dynamically and precisely reflect the evolution of users’ characteristics (which is especially important for AES). Among the drawbacks of this approach is the necessity of developing an accurate and formal domain model, which is a hard task for some domains.

### 2.2.2 Keyword-based user modelling

Keyword-based user modelling originated in the areas of information retrieval and filtering (Belkin and Croft, 1992), where the content of a document is traditionally represented as a vector of terms (keywords) extracted from the text. The adaptive information retrieval and filtering applications aggregate a user’s history of launched queries, accessed documents, or rejected e-mails in a form of a keyword vector and use this vector for tailoring a future retrieval or filtering process to the user’s information-seeking idiosyncrasies.

To a certain extent, this approach can be considered a shallow version of overlay user modelling. It also utilises elements of domain representation as a frame of reference to express user characteristics on the atomic level. However, instead of concepts modelling domain semantics, this technique uses keywords/terms found in the content. It became very popular in the context of adaptive information retrieval on the web (Brusilovsky and Tasso, 2004). Many AW systems model users’ interests or needs as vectors of keywords extracted from the documents the users have browsed or requested, as for example in Letizia (Lieberman, 1995), WebMate (Chen and Sycara, 1998), NewsDude (Billsus and Pazzani, 1999), etc. Some of the systems model users’ interests as networks of keywords instead of plain lists, where nodes represent keywords and arcs connect keywords co-occurring in the content, as in iWeb (Asnicar and Tasso, 1997) and PIN (Tan and Teo, 1998).

A big advantage of this approach is the automatic modelling of content based on well-developed IR techniques for text analysis, which also opens opportunities for open-corpus adaptation. However, keywords support only shallow content models. To remedy problems like homonymy (multiple meanings of a word) and synonymy (multiple words expressing the same meaning), Natural Language (NL) technologies are required. Pure keyword-based modelling is not able to represent the true meaning of the content. It relies on statistical regularities within the text and, essentially, provides a framework for retrieving statistically close documents.

### 2.2.3 Stereotype user modelling

The ultimate goal of adaptive systems is to best adjust its behaviour to the characteristics of individual users. However, for some tasks it is possible to identify typical categories of users that use the system in a similar way, expect from it similar outcomes and can be described by similar sets of features. Such categories “constituting strong points of commonalities” (Kay, 1994) among users are called stereotypes.

An adaptive system relying on stereotype-based modelling does not update every single facet of the user model directly. Instead, it utilises a stock of preset stereotype profiles. Whenever the system receives an evidence of a user being characterised by a certain stereotype, the entire user model is updated with the information from that stereotype profile. A user can be described by one stereotype or a combination of several orthogonal stereotypes. A popular way of stereotype-based user modelling is a linear set of categories for representing typical levels of user proficiency. For example, Chin (1989) describes the KNOME system modelling users’ expertise of the UNIX operating system on one of four levels (novice, beginner, intermediate, and expert). Often, however, stereotypes form a hierarchy, where more specific stereotypes can inherit some information from their parents, as in the classic system Grundy (Rich, 1979).

Stereotype-based user modelling is advantageous when from a little evidence about a user the system should infer a great deal of modelling information. However, for modelling fine-grained characteristics of individual users (for example, a knowledge level of a particular concept) more precise overlay models should be employed.

### 2.2.4 Other technologies for user model representation

We have described these three approaches in details due to the many projects employing the synergy between them and the web ontologies. However, other important technologies for user model representation should be mentioned as well.

Constrained-based user modelling has been successfully used for modelling student’s knowledge in Intelligent Tutoring Systems (ITSs) (Mitrovic et al., 2001). In this approach every constraint represents an acceptable set of
equivalent problem states and a violated constraint indicates an error.

One of the dominating adaptive technologies on the web nowadays is collaborative filtering. Unlike most of the user modelling technologies mentioned before it relies on modelling users in terms of their relationships with other users. The typical collaborative user model is based on a vector of ratings the user provided for particular items (Konstan et al., 1997).

Bayesian Networks is yet another very popular formalism for representing different aspects of user models. Multiple systems used Bayesian Networks to model the relations between different components/dimensions of a user model, such as emotions, goals and knowledge (e.g., Zhou and Conati, 2003). Other systems used them to implement an overlay user model with internal inference capabilities, where every node represents a domain concept and links stay for the concept relations, e.g., de Rosis et al. (1992).

In the e-commerce applications it is often effective to model a customer without deriving any explicit modelling assumptions about him, but rather by identifying certain statistical regularities that can be utilised for building effective selling strategies (Agrawal et al., 1993). A user model in this case can contain a filtered set of transactions matched against an association rule of items bought together or satisfying some linear pattern of buyers’ behaviour, or belonging to a cluster of similar buyers. Recently this approach has been employed for modelling web users as well (Baumgarten et al., 2000).

2.3 User model elicitation

There are two principle ways for an adaptive system to obtain user modelling information: to ask the user directly or to derive it based on the user’s activity with the system. Many systems use a combination of these approaches, as the natural flow of the ‘user-system’ communication (defined by the task the user tries to complete) often requires active user feedback. For example, a student, solving problems in an ITS, has to provide intermediate and final answers, based on which the system can infer their knowledge of target concepts. Other systems, however, rely only on their ability to mine information about the users from the logs of their actions by applying machine learning techniques. If the main task of a user does not imply any direct input to the system, such an approach is preferable, as it does not interfere with the task-related activity and does not increase the user’s cognitive load. Finally, some adaptive systems actively involve the user in the modelling process. They may allow or even encourage users to directly modify their user models. This section describes briefly these elicitation technologies.

2.3.1 User model inference based on rich feedback

The most typical examples of adaptive systems receiving rich feedback from users come from the broad class of NLP systems. In intelligent dialog systems user modelling has been summoned to help in recognising the goals/plans/beliefs of the user and to provide a basis for a more justified and robust dialog actions. Typically, the interface of a dialog system is text-based, where a user is able to enter his utterances in free manner, either asking the system a question, or giving an answer to the system’s qualifying question. Both users’ answers and questions are used to gradually construct the user model; the system’s responses are often driven by the ‘intention’ to refine some of the modelling assumptions. For example, in Grundy (Rich, 1979), the system plays the role of a librarian helping the user to choose an appropriate book.

To elicit individual user models, intelligent dialog systems of the early user modelling age usually followed one of two ways: they either attempted to ask the user a fishing question, or utilised rules, predicate logic and other classic AI reasoning techniques, especially, such used for plan recognition. Many of the developed reasoning components and ad-hoc heuristics were domain-dependant and could not be generalised for other tasks. The most detailed overview of the early works in this area is provided by Wahlster and Kobsa (1989). Zukerman and Litman (2001) analysed some of the more recent efforts to employ user modelling for NLP applications.

The second large class of adaptive systems relying on both the rich user feedback and their inference capabilities is constituted by various AESs and especially by ITSs. The developers of ITS extensively use technologies for reasoning under uncertainty. A typical user input in an ITS is providing an answer to a problem or taking a step towards the problem solution. Based on this information, as well as on the domain model and the problem description, the ITS employs its reasoning mechanism to update the user model, which will be later used in certain adaptive actions (presenting a problem of optimal difficulty, generating remedial hints, suggesting a review of a tutorial, etc).

Various approaches have been applied for deducing user characteristics. The mechanism known as knowledge tracing uses simple Bayesian inference to calculate the probability that the user has mastered a certain skill based on the prior probability for that skill and the evidence provided by the user (a correct/incorrect solution of the corresponding problem) (Corbett and Anderson, 1994; Galeev et al., 1994). A generalisation of this methodology is employed by several systems using Bayesian Networks (e.g., Jameson, 1992). The Bayesian Network in Epi-Umod (de Rosis et al., 1992) models user knowledge as an overlay over the network of domain concepts. Conditional probabilities represent the mutual effects of knowledge of relevant concepts; hence, the propagation of user knowledge over this network takes into account the prerequisite-outcome relationships between them.

In addition to Bayesian inference, some other technologies have been also used for this purpose, such as the Dempster-Shafer’s approach (Petrushin and Sinitza, 1993), Hidden Markov models (Beal et al., 2007), or Fuzzy Logic (Capuano et al., 2000).
2.3.2 Unobtrusive user modelling

In order to support users in the most efficient way and reduce their cognitive load, an adaptive system should try to minimise the interference with the main task performed by the user needed to maintain the adaptation. The exponential growth of information and users on the web, as well as the transfer of many services and activities to electronic and online forms, call for applications that support users unobtrusively by navigating them to the desired information or filter out the non-relevant web pages, documents, products, etc.

Webb et al. (2001) specify the challenges of unobtrusive user modelling based on machine learning techniques. The amount of information collected by an adaptive application about their users is often not enough to build a straightforward computational model of acceptable accuracy (small datasets). Generally, user modelling is a dynamic task; hence the parameters characterising a user are likely to change over time, however, ML modelling algorithms are often not able to adjust to these changes quickly enough (concept drift). User modelling applications are supposed to operate in an interactive mode, however, many of the ML algorithms take a long time to converge (interactivity vs. computational complexity). Supervised ML algorithms require explicit labels. In the context of user modelling, such labels often can be provided only by the users (for example, the user notifies the system that she or he does or does not like this webpage). Unfortunately, users are known for being unwilling to provide information, which is not directly connected to their needs (lack of labelled data).

Pierrakos et al. (2003) and Mobasher (2007) provide a detailed description of the main stages of information flow in these applications as well as the techniques they use on every stage. There are three main stages common for all data mining systems: data collection, data pre-processing and pattern discovery. Some applications perform the last two stages in offline mode.

On the first stage, data from different sources is collected, combined and structured to provide the basis for the following processing. The main source of data is provided by the web server’s transactional logs, which store the basic usage data, the click stream of users accessing system’s resources (date/time, requesting IP address, requested resource, HTTP method, user platform characteristics, etc.).

Data pre-processing involves several tasks. First, the raw click stream data should be cleaned from the references to non-informational resources (graphics, audio, style sheets, etc.) and transactions generated by spiders. Then the missing (due to the caching) transactions are generated based on the analysis of the site hyper-structure. Other important tasks performed on this stage include user identification and session separation. This step is very important, since every session is characterised by a specific information task and represents a logically complete pattern of user’s navigational behaviour.

The final stage, which is most related to user modelling, is pattern discovery. On this stage, the application mines from the processed and enriched transactional data statistical patterns representing observed usage regularities, which are later utilised for recommending an appropriate resource, navigating towards a relevant web page or retrieving a document with desired features. There are several groups of mining techniques coming from the field of ML: clustering, classification, association rules, sequential patterns, and latent variable models.

Clustering methods have been used mainly for two purposes. User sessions have been clustered to mine the typical navigation patterns. As a result, the systems were able to recommend hyperlinks based on the other sessions in the cluster (Yan et al., 1996), to generate an index page navigating the user to all pages from the current session’s cluster (Perkowitz and Etzioni, 2000), etc. Collaborative filtering systems often use offline clustering of users and/or items to remedy the scalability problem (Mobasher, 2007).

Classification has been less popular than clustering among the adaptive data mining applications, because of the need of pre-classified (labelled) data, which is rarely available. Most often, classification has been used for building descriptive models of user interests. For example, when a user browsing on the web chooses to save, print or bookmark a certain page, a classification component can take it as a good indicator of the user’s interest in the content of this web page. The pioneer Letizia system uses this approach to recommend links potentially interesting to the user (Lieberman, 1995).

Association rules and sequential patterns are very similar and both have been developed as techniques for market basket analysis. The difference between them is based on whether the order of transactions is taken into account by the algorithm or not. While sequential patterns utilise this information, association rules containing the same items but in different order are considered identical. Similar to clustering, these techniques are used to predict the possible navigational path of the user and support him by recommending a proper shortcut (Mobasher et al., 1999).

Several latent variable models have been employed by the adaptive data mining applications (see, Mobasher, 2007) for a comprehensive review). Their common premise is to model some hidden (unobservable) factors influencing observable user behaviour by introducing a latent variable (or a set of variables). Latent variable models successfully apply probabilistic inference to uncover semantic relationships hidden in the data. For example, Jin et al. (2005) used probabilistic latent semantic analysis to predict information tasks of a user with a set of latent parameters based solely on the navigational patterns. Pierrakos and Palouras (2005) exploited similar technique to uncover hidden motives of web users and on that basis to model online user communities.

2.3.3 Open, editable and interactive user modelling

In 1988, while discussing problems of intractability of student modelling, Self (1988) suggested an idea opposite to the traditional user modelling approaches described in the previous sections: “Avoid guessing – get the student to tell you what you need to know”. He suggested to
“make the contents of the student model open to the student, in order to provoke the student to reflect upon its contents and to remove all pretence that the ITS has a perfect understanding of the student…” (Self, 1988)

Since then, a number of adaptive systems have explored the effects of opening the user model and providing the user with different levels of control over it (inspecting, interacting, scrutinising, modifying). Most of these systems are AESs, so, it is more common to talk about open learner modelling (Kay, 2001).

The first implementation of this idea was the tool named skillometer (Corbett and Anderson, 1992) (see Figure 1). It contains a list of skills that a student is to achieve with the system’s help and the current state of proficiency for each skill, modelled by the system. Skillometers provide an example of both: the simplest visualisation of the user model and the most basic mechanism of ‘user-model’ interaction (users are informed about their current learning state). More complex interfaces have been developed to advance the idea of open user modelling in both directions. For example, Zapata-Rivera and Greer (2004) presents ViSMod, a system supporting visualisation of Bayesian learner models (see Figure 2). The visualisation presents nodes and relations between them, and prior and conditional probabilities; it allows the user to set up evidences and explore the model in ‘what-if’ scenarios.

Figure 1  
Skillometer – the basic open learner model

Flexi-OLM explores further several elaborate techniques for presenting learner models:

“hierarchy, a logical grouping of related concepts; lectures, where topics are organised the same as in the related lecture course; a concept map showing relationships between the topics; prerequisites, showing possible sequences for studying topics; index, an alphabetical list; ranked, where topics are listed in order of proficiency; and a textual summary.” (Mabbott and Bull, 2004)

It also provides students with multiple functionalities for communicating with the model. A student can browse the model and reflect upon its content; she or he can directly edit the model, by providing the self-estimation of the knowledge level for a particular concept; if not certain about his or her knowledge a student can ask the system about the reason for their grade and try to persuade it by answering a series of test questions (Mabbott and Bull, 2006).

Figure 2  
Open Bayesian learner model (see online version for colours)

Another example of an open learner model requiring active participation of the learner in the modelling process is provided by Dimitrova (2003). The STyLE-OLM system implements a platform, where the conceptual model of learner knowledge is built interactively, through negotiation between the learner and the system.

Scrubtable user modelling provides a user with an access to the internal system’s modelling assumptions (Kay, 1999, 2006). A user not only can inspect the current state of the system’s beliefs about her or him, but is also capable to ‘scrutinise’ the model and get an answer to the questions like: “What does it mean?”, “How well do I know this topic?”, “Why my knowledge level is X?”, etc.

Kay (1999) demonstrates the usefulness of scrutable user modelling in a text editor help system and adaptive movie recommender. Ahn et al. (2007) presents the YourNews system, investigating the feasibility of open and editable keyword-based user profiles for adaptive news recommendation.

Figure 3 shows the interface of YourNews; users can observe their profiles, built by the system, remove some of the keywords from the profile (crossed words), or add a new keyword to improve the recommendation.

Figure 3  
YourNews: editable user profile for adaptive news recommendation (see online version for colours)
The preliminary results demonstrate that unlike the concept-based editable user models, keyword-based models are not edited by users to their advantage. This fact can be explained by the shallow nature of keyword-based user profiles. A concept-based domain model consists of elements providing users with comprehensible semantic information; therefore concepts can successfully serve as a basis for the open and editable user modelling. On the contrary, the meaning of automatically populated keyword vectors is based on statistical regularities found in the content. A single keyword does not represent a solid piece of information by itself; therefore keywords cannot ensure consistent editable user modelling. A blending of these approaches is a perspective direction of research. A system capable of translation between keyword-based and concept-based models would benefit of both: the automatic modelling of open-corpus data and semantically rich user profiles.

3 Ontologies

Since Tim Berners-Lee put online his SW Road map (Berners-Lee, 1998) and Scientific American published the seminal SW paper (Berners-Lee et al., 2001), the main idea of SW has not changed much. The SW is seen as a web of meaningfully described information, available for automatic discovery and integration across distributed applications. However, it has been recognised by now that the time needed for the realisation of this idea will be much longer (Shadbolt et al., 2006). The amount of available RDF-based metadata stays very small compared to the volume of the existing information on the web. The broad reuse and sharing of web data on the basis of robust and commonly accepted ontologies, as well as the large-scale agent-based mediation of web services, are not happening yet.

Nevertheless, there are local stories of success. Some companies are already developing SW applications (Radar Networks, 2008; Reuters, 2008; SemantiNet, 2008; Zemanta, 2008; Benjamins, 2006; iSOCO, 1999). The number of researchers working on SW problems is constantly growing. Several closed communities have become enough motivated and well organised to create small ‘islands’ of SW (Health Care and Life Sciences (W3C, 2007), Public Administration (SWED, 2004), Engineering (SWOP, 2005), etc.). Overall, the SW initiative has resulted in the development of a number of useful technologies. Many of them are related to different aspects of KR employing web ontologies. Although ontologies are not a necessary element for any SW application, if the task of the application goes beyond the integration of simple web data and requires the representation of knowledge on the web, ontologies provide the mechanism to do so.

3.1 Ontology representation and development

3.1.1 RDF and RDF-schema

Resource Description Framework (RDF) (Lassila and Swick, 1999) has been proposed by W3C as the major mechanism for metadata representation on the web. The RDF model is similar to the Semantic Network formalism and could be visualised as a directed graph. The elementary particle of the RDF model is a resource representing anything that can be referred by URI. To uncover a resource’s semantics, one should link it to other resources. RDF statements provide the mechanism for organising resources into triples “subject-predicate-object”. Every triple represents a directed semantic link from one resource to another. As a result, interconnected resources organise a semantic network.

The original RDF vocabulary is limited to the basic set of primitives for managing descriptive metadata. To create richer RDF documents, one needs more advanced tools for developing collections of RDF terms enriched with interpretable semantics. For these purposes W3C has proposed the RDF vocabulary description language – RDF Schema (RDFS) (Brickley and Guha, 1999, 2004). RDFS provides two important extensions for the RDF model. First, it specifies the mechanism for defining classes of resources and developing hierarchies of classes and properties. Second, it allows the restriction of property applicability to the resources from a particular class. In contrast to pure RDF, which can be used solely for descriptive metadata specification (e.g., “who is the author of the resource?”) RDFS allows the design of lightweight domain ontologies and specification of resource semantics (“what is the resource about?”) in terms of such ontologies.

3.1.2 The web ontology language

Although RDFS can be used as a representation formalism for web ontologies, its expressiveness does not allow the creation of elaborate domain models beyond simple class hierarchies. If the task requires the creation of a heavy-weight ontology modelling deep-level domain semantics, a more complex ontology language should be used. Well-defined semantics is necessary for supporting machine interpretability, as it allows avoiding potential ambiguities in the description interpretations. Hence, the higher the formality of the ontology language, the more complex reasoning it allows and the more useful the developed ontologies can be.

In 2004 W3C recommended the Web Ontology Language (OWL) as the major formalism for ontology representation on the SW (McGuinness and van Harmelen, 2004). OWL was developed as a revision of the DAML+OIL ontology language (Connolly et al., 2001) and inherited most of its features. The consistency of OWL with the lower-level web standards has been preserved. OWL is based on the generic RDF model; it uses RDF/XML syntax and the RDFS terms are naturally integrated in the OWL vocabulary. At the same time, OWL introduces a number of new features to handle semantically complete metadata models. The creators of OWL built the language upon such rich representational formalisms as frame theory (Minsky, 1974) and description logics (Baader et al., 2003) (for a detailed analysis of the origins of OWL semantics (see Horrocks et al., 2003). As a result,
OWL allows modelling of all the major components of formal ontologies: classes (or concepts), individuals (or instances), properties (including class relations and instance attributes), and axioms (or constraints).

There are three versions of OWL and each provides ontology developers with more expressiveness than the previous one: OWL-Lite, having a very limited set of features beyond basic class hierarchies; OWL-DL, enhancing OWL-Lite with the functionality of description logic; and OWL-Full, combining the expressiveness of the OWL-DL vocabulary and the freedom of the RDF model.

3.1.3 Other semantic web representation technologies

Although the basic metadata representation technologies have been introduced a while ago, there still exist a number of well-recognised metadata problems slowing down the emerging of the SW. As a potential answer for these problems, several new RDF-based technologies are currently undergoing standardisation in W3C.

One of the biggest problems of the SW is the simple lack of metadata-enriched web content. For spreading the semantic technologies some critical mass of RDF-annotated content has to be accumulated. The true power of the web was revealed only when enough information appeared in HTML format and the separate HTML pages became connected to each other. The same effect should eventually bring SW applications from the research labs to wide use by the general public. However, currently the volume of SW metadata is very far from web-like ubiquity. According to Pandia (2007) the size of the web in February 2007 was somewhat between 15 and 30 billion of documents. Swoogle (Semantic Web Search Engine) (Ding et al., 2004) can recognise currently 2.2 million RDF documents (Swoogle, 2007), which is only about 0.01% of the overall web content.

To remedy the insufficiency of RDF-based metadata on the web, W3C proposes several solutions. One of them is provided by Gleaning Resource Descriptions from Dialects of Languages (GRDDL) (Connolly, 2007). GRDDL is a new technology specifying a formal way to retrieve RDF content from XHTML pages and multiple XML-based microformats existing on the web. XML Transformation languages such as XSLT can be used to translate between different XML dialects like XHTML and RDF/XML.

Another technology trying to address the same problem but from a different angle is RDFa (that, arguably, stays for RDF-Attributes) (Adida and Birbeck, 2007). RDFa follows the examples of such technologies as SHOE (Luke et al., 1996) and Ontobroker (Decker et al., 1998). Instead of converting the existing HTML pages into RDF, it provides a mark-up extension for incorporating RDF-based metadata attributes into HTML pages.

The lack of SW metadata is partly related to the lack of well-accepted RDF-based vocabularies. There are successful examples of widely-used collections of RDF-terms, such as Dublin Core for digital libraries (DCMI, 1999), and FOAF for social networks (Brickley and Miller, 2005), however they cannot support the metadata creation for every application task. Simple Knowledge Organisation System (SKOS) is aimed to provide a general model and a core vocabulary for representing such knowledge structures as thesauri, glossaries, taxonomies, folksonomies, etc. The main objective of SKOS is to support easy development and publication of controlled vocabularies for the SW (Miles and Brickley, 2007).

3.1.4 Ontology development tools

The availability of well-accepted standard technologies for ontology representation, based on formal models (RDF graph model, Description Logic model) and machine-readable syntax (XML), greatly facilitates ontology development. Before the introduction of the RDF-based representational models, no common way for sharing developed ontologies existed; the ontology authoring tools used their own formats, e.g., Ontolingua (Gruber, 1993) used KIF and OKBC, Ontobroker (Decker et al., 1998) used F-Logic. For some previous XML-based ontology languages like XOL (Karp et al., 1999) and OML (Karp et al., 1999), it was possible to use general XML editors. The RDF-based ontology languages allow an ontology author working with one development tool to open, explore and reuse ontologies developed by other authors in other tools and to share their ontologies.

Multiple tools for ontology development have been recently created and multiple reviewing papers comparing different aspects of these systems have been published (e.g., Gomez-Perez et al., 2002; Polikoff, 2003; Youn and McLeod, 2006) provide an aggregated review of some two dozens ontology development tools).

The most famous and popular line of tools for ontology development is the Protégé product family, developed at the Stanford University School of Medicine (Noy et al., 2001). Protégé has two versions: Protégé-OWL, for creating SW ontologies, and Protégé-Frames, for developing knowledge models compatible with the OKBC framework. The standard functionality of Protégé contains an extensive set of features to work with basic elements of OWL (classes, properties, individuals and constraints), however, the real power of Protégé comes from its extensible architecture. Protégé API allows a third-party’s plug-ins to be easily integrated into the system and extend it with advanced functionality for visualisation, import/export, reasoning, querying, mapping, and integration with other tools (the number of registered plug-ins at the moment of writing this manuscript is 87). Another important feature of Protégé, distinguishing it from most of the other ontology development tools, is scalability.

3.2 Ontology mapping

Ontologies are intended to represent consensual knowledge in the domain of discourse. Ideally, a domain ontology should be discussed and verified by all interested comminutes. Unfortunately, achieving such a level of
agreement is unrealistic in the web settings. Most often, related communities differ in their perspectives on the same domain in such a degree that common ontological commitment across all interested parties is impossible. However, from a practical point of view, this should not be even required, as an ontology representing a community’s perspective provides a unified-enough model that can effectively serve as a basis for semantics sharing and data integration.

As a result, there are many ontologies representing the same domain and their number will continue to grow as the SW ideas disseminate further. Even larger is the number of overlapping ontologies (ontologies modelling the same concept(s)). For example, the Swoogle search engine (Ding et al., 2004) currently retrieves 3900 RDF documents referring to the resource named ‘user’. Some of these documents are actual ontologies defining their own versions of the concept ‘user’ reflecting different viewpoints and notions but still modelling the same (or very close) entities in the world. This number becomes even greater if we consider ontologies defining the same or very close concept using different labels (e.g., ‘actor’ or ‘human agent’).

To find the correspondence between identical and relevant concepts defined in different ontologies, various ontology integration techniques have been developed over the last ten years. These techniques have formed, arguably, the most dynamic area of ontological research named ‘ontology mapping’.2 Dozens of theoretical frameworks, formal algorithms and actual systems for ontology mapping have been created and several comprehensive reviews have been written (see for example, Kalfoglou and Schorelmmer, 2003; Noy, 2004).

### 3.2.1 Ontology mapping based on a reference ontology

In a limited number of cases, the ontologies being mapped can contain references to a common ontology. Most often, the common ontology models very general notions of nature, such as time, space, process etc. Several initiatives to create general ontologies exist: Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2001), Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Gangemi et al., 2002), CYC (Lenat, 1995), WordNet (Miller et al., 1990) etc.

The natural purpose of a general ontology is to be extended by domain-specific ontologies. If concepts of two ontologies are connected to concepts of an upper-level ontology, such linkage serves as a valuable source for identification of correspondences between the mapped ontologies. For example, if two ontologies have their concepts ‘actor’ and ‘human agent’ linked with ‘sameAs’ relations to the single concept ‘user’ of the upper ontology, this provides strong evidence that these two concepts from different ontologies actually model the same entity in the world.

### 3.2.2 Ontology mapping based on lexical information

Probably the most valuable information for ontology mapping comes from the lexical labels given to the elements of the ontologies by their developers. Nicely designed ontologies also contain natural language comments and definitions explaining the meaning of the classes, predicates and axioms constituting the ontology. Ontology developers tend to maintain this lexical information meaningful and consistent across the ontology, therefore lexical similarity of two elements often indicates the existence of semantic similarity.

Utilisation of lexical information is traditional for ontology mapping techniques. Even if the system applies other mapping methods, it somehow exploits the lexical information as well. For example, Hovy (1998) describes a method for semi-automatic alignment of domain ontologies to a central ontology. The method is based on a set of heuristics for lexical comparison of concept labels and definitions across the ontologies. The lexical component of the method first breaks composite names into words, then analyses word sets for common substrings and computes the ratio of common words. Finally, a limited analysis of taxonomical relations among the concepts is performed. The GLUE algorithm (Doan et al., 2002) combines the results of two separate mappers, one of which is based on the lexical similarity of extended class names (where an extended class name is a concatenation of a class name and names of all its super-classes up to the root class of the ontology).

### 3.2.3 Ontology mapping based on ontology structure

An ontology represents the objective relationships among the knowledge items of the domain. Therefore, if two ontologies of the same domain are designed in a meaningful way, the parts modelling the same piece of domain knowledge should have similar structures in the ontology graphs. Thus, another source of evidence that can signify the alignment among elements of two ontologies is the resembling structural patterns in the ontology graphs.

Most of the ontology mapping techniques rely on structural relationships among ontology elements to a certain degree. The simplest reasoning pattern would be similar to the following: if two concepts of the ontologies are mapped, their super-classes are likely to map, same as their subclasses, same as their siblings. Similar heuristics are implemented in the PROMPT ontology mapping algorithm (Noy and Musen, 2003). Once it finds a confirmed mapping, it starts harvesting the neighbourhoods of the mapped elements to suggest new mappings. PROMPT’s later modification AnchorPROMPT analyses ontologies as directed labelled graphs, where nodes represent classes and links – predicates. AnchorPROMPT considers two sub-graphs from the ontologies and compares possible paths through the subgroups restricted by anchors.
If, as a result of the analysis, two classes often occur in the same parts of the graphs, they represent similar domain concepts.

3.2.4 Ontology mapping based on instance corpora

Some ontology mapping algorithms try to utilise not only the content of ontologies themselves, but also the corpora of textual instances annotated with the ontology elements. An access to sets of instances significantly increases the amount of data available for the ontology mapping algorithm. To manage such volumes of information effectively, ontology-mapping algorithms implement various probabilistic and machine learning methods. This allows them to benefit of well-developed methodology for statistical reasoning based on unstructured textual data.

One example of such algorithms is implemented in the project Formal Concept Analysis for bottom-up ontology merging. The combined set of class instances of the original ontologies undergoes lexical preprocessing and then is used to create the merged concept lattice of the two ontologies. After the resulting lattice is automatically pruned, a human expert should analyse it and manually extract from it the resulting ontology.

The GLUE algorithm (Doan et al., 2002) is another example of machine learning-based ontology mapping. Its instance-based mapping component takes as input two ontologies along with the corresponding sets of instances. On the first step, for every pair of concepts from the two ontologies GLUE estimates the joint probability of their distribution in the joint set of instances. For this purpose, GLUE trains a classifier for every concept using its instance set as positive (instance is annotated with the concept) and negative (instance is not annotated with the concept) cases and then cross-classifies instances of the second ontology using the just-trained classifiers. Then GLUE repeats the procedure with the ontology roles reversed. Thus, for every pair of concepts GLUE is able to break the joint set of instances into four subsets (instance is classified positively for both concepts, for one of them, or for none them). The obtained joint probability distributions are converted into concept similarity estimations. As a final step, for every concept GLUE chooses the mapping candidate from the other ontology, for which it has the maximum similarity value.

3.3 Ontology visualisation

Most of the AI and SW research refers to ontologies as the means for conveying semantics to machines and helping machines to understand each other. However, the potential use of ontology in a knowledge-based application is not reduced to interfacing programs; a user interface of a system can also exploit ontologies for effective navigation of a user through the content (Fluit et al., 2005; Guarino, 1998).

Such an interface must present to the users the elements of the underlying ontology in a meaningful and useful way.

The difference of ontology visualisation from the general visualisation of a structured unit of information spreads from the exceptionally rich semantics of ontologies. Ontology visualisers should provide extra capabilities for explicating this semantics (Alani, 2003). Effective ontology visualisation could be helpful in a number of scenarios, which can be reduced to two general tasks:

- visualisation of the ontology itself for the purposes of ontological analysis and understanding of the domain semantics
- ontology-based information visualisation for effective navigation (e.g., ontology-based search engine could present the results of a user’s query not as a flat list of hits, but as an ontology-based structure reflecting the semantics of the hits (Fluit et al., 2005)).

3.3.1 Visualisation of ontologies

Every ontology editor has some ontology visualisation capabilities to assist ontology designers in their job (e.g., to observe the results of current changes, validate design decisions and check for inconsistencies and mistakes). Typically, an ontology editor presents an ontology as a class hierarchy. Class hierarchies provide ontology developers with simple navigational capabilities. They facilitate the definition of class-subclass relations and help to focus on a certain class for defining its predicates and instances.

However, supporting ontology developers is not the only goal of ontology visualisation. A nicely visualised ontology could be used as a powerful cognitive tool. The ontology itself provides valuable information describing domain semantics in a structured and unbiased way. Comprehensible visualisation of an ontology can help the user to understand and analyse information on the conceptual level.

To meet these requirements some of the ontological tools support more advanced visualisation layouts. The best example here is Protégé, whose plug-in architecture accommodates a dozen of visualisation tools, including (the two most popular) TGVizTab (Alani, 2003) and Jambalaya (Lintern and Storey, 2005; Storey et al., 2001).

3.3.2 Ontology-based information visualisation

Several recent projects apply visualisation techniques for implementing ontology-driven access to large amounts of information. Although ontology visualisers mentioned above can also be used for providing access to class instances (that can carry valuable information), there is a major difference between them and the projects described here. The previous section focused on the visualisation of an ontology itself that supports a user in conceptual knowledge analysis and comprehension of domain semantics. This section concentrates on designing interfaces for meaningful
information access based on the domain semantics that facilitates such activities as efficient information seeking and navigation.

An example of a project designed to support users’ exploratory activities is **Spectacle** (Fluit et al., 2005). It provides a convenient interface for navigating through a corpus of structured information and finding interesting informational resources. **Spectacle** introduces a novel visualisation mechanism called Cluster Maps. Cluster Maps are designed for presenting information annotated in terms of lightweight ontologies as networks of clusters (see Figure 4). Every cluster represents a set of information resources attached to a concept annotating these resources. If a set of resources belongs to a number of concepts, they form a cluster attached to all the concepts involved. Every small sphere in a cluster’s bubble represents an actual single resource. The concepts themselves are connected with directed links denoting a hierarchical relation. A click on a concept node makes this node a focal concept; it opens the neighbouring concepts along with their resource cluster and converge the irrelevant concepts that stay too far from the focal node in the hierarchy. If the corpus of resources is annotated using different categories, a user can browse the same information from different perspectives.

**Figure 4** Visualisation of job vacancies organised by economy sector in **Spectacle** (see online version for colours)

---

### 3.4 Ontology-based knowledge acquisition

The knowledge acquisition bottleneck remains the main problem for the SW technologies dissemination. Although new technologies for developing metadata vocabularies (e.g., SKOS) and simplified embedding of RDF into HTML content (e.g., RDFa) have been introduced, the large-scale manual development of semantic metadata stays rather an unrealistic scenario due to a number of reasons:

- manual authoring of metadata is time-consuming and expensive
- manual authoring of metadata is difficult and error prone
- manually created metadata is often subjective (domain experts developing a metadata vocabulary for the same field or annotating the same documents are likely to disagree on a large portion of the work)
- metadata creation is never ending (new documents appear, new versions of ontologies are developed, new fields and tasks arise)
- metadata storage and ownership problem (where the manually created metadata should be stored, who is responsible for its support, who owns it?)
- metadata created manually by the owner of a document may be not trustworthy (he can be incompetent or malicious, e.g., for spam reasons).

These issues highlight the importance of developing online metadata services (similar to the modern web search engines) relying on technologies for automatic semantic metadata creation. Two research subfields try to employ ontologies in this process. Ontology-based annotation combines a set of techniques for automatic and semi-automatic description of web documents with semantic metadata. Ontology learning studies methods for building ontologies from scratch or enriching existing ontologies.

#### 3.4.1 Ontology-based annotation

Annotation of documents with semantic metadata can be used for different purposes:

- to support integration of disperse information in the document space (knowledge-based implicit linking of documents vs. traditional explicit linking)
- to ensure better indexing and retrieval of documents (based on the document semantics vs. shallow keyword-based content modelling)
- to support content-based adaptation (by modelling document content in terms of the domain model).

Ontology-based annotation can be defined as a process of creating a mark-up of web resources using a pre-existing ontology and/or populating knowledge bases via the mediation of marked up documents (Gomez-Perez et al., 2002). Figure 5 visualises the typical process of ontology-based text annotation. Some of the terms in the text are matched against the elements of the ontology. As a result, the knowledge-based system becomes aware, that this extract speaks about an entity **XYZ** and entity **Bulgaria**. The ontology reveals also that **Bulgaria** is a country, while **XYZ** is a company located in **London**, **UK**; where **London** is a city, and **UK** is a country, different from **Bulgaria**.
Modern ontology-based annotation systems provide manual, semi-automatic or fully automatic annotation. The ontologies used for annotation can be represented in different languages. Some systems annotate web resources only in terms of a specific ontology, while others allow loading external ontologies. The resulting annotation can be represented in RDF, XML or other formats; it can be attached to the annotated documents or stored separately on a dedicated server. Some systems use for annotation only class names, while others allow annotating with relations, instances and even entire triples. The architecture of a tool can vary from an API to a client-server application. Finally, the scale of annotation is a very important issue; while some tools aim at annotating relatively small sets of documents, others provide large-scale annotation of millions of resources.

Semi-automatic ontology-based annotation

Semi-automatic (or user-centred) ontology-based annotation is the current mainstream approach for semantic markup of web documents. Its main idea is based on the traditional machine learning methods for text classification.

While a human annotator creates a markup for initial limited set of documents, the information extraction component (such as Amilcare (Ciravegna, 2001)) learns the associations between content features and ontology elements. When ready, the algorithm continues the annotation in automatic fashion. At the end, the human annotator usually has an option to review the results of the automatic phase and make necessary corrections.

The tool functionality and user interface vary from one system to another. Some systems provide extra features. For example, MnM can work with several information extraction components performing the automatic phase of annotation (Vargas-Vera et al., 2002). It can access ontologies stored on external ontology servers (such as OKBC) through APIs and stores annotations on a dedicated server. Another semi-automatic annotation system Melita (Ciravegna et al., 2002) organises the annotation process in an adaptive manner. It does not wait until the end of the training phase, instead, it starts recommending annotation candidates immediately after the human annotator has provided some evidence.

S-CREAM (Henschuh and Staab, 2003; Henschuh et al., 2002) can crawl previously annotated web resources and use them as training data. It also supports annotation of the so-called deep web (web content dynamically created from relational databases thus not accessible for retrieval and annotation by traditional tools) (Bergman, 2001). Some experts estimate the proportion of the deep web with respect to the shallow (static) web as 500 to 1. To annotate such dynamic content, S-CREAM analyses the schemas of underlying databases. It considers these schemas as ontologies and applies ontology mapping algorithms to identify the correspondence between the schemas and the target ontology. The resulting mapping rules are used to dynamically markup the generated documents.

Fully automatic ontology-based annotation

Unlike semi-automatic ontology-based annotation, fully automatic approaches considerably differ one from another. The algorithms employed are often based on ad-hoc hypotheses rather than on a well-accepted methodology, such as text classification. The common feature of these approaches is the wide usage of NLP techniques. However, the complexities of the employed NLP algorithms are very different. Some systems, like COHSE (Bechhofer et al., 2003), do not go beyond the identification of noun phrases in the text and a trivial match of the found entities against concepts from an ontology enriched with synonym information from a thesaurus. Others, like AeroDAML (Kogut and Holmes, 2001), in addition apply predefined linguistic patterns to identify instances of typical categories (date, time, money, etc.). Consequently, the first phase of most algorithms for automatic annotation is usually the same – the textual corpus is analysed for words or phrases conveying the content semantics. The difference is in choosing annotation candidates to map a piece of content to the appropriate semantic tag from the ontology.

For example, the PANKOW algorithm (Cimiano et al., 2004) attempts to rely in this process on the collective intelligence harvested from Google users. After PANKOW chooses candidate proper nouns that can be important for the corpus, it builds a set of hypothesis queries from the found candidate nouns and candidate annotating concepts for these nouns. The queries are created using predefined linguistic patterns. On the next step PANKOW tosses the generated queries to the Google search engine (using Google API). The results retrieved from Google are compared based on the numbers of hits for every hypothesis.
query. The noun-concept pair, which formed the set of queries with the highest cumulative hit rating is considered the most related, hence that annotation is performed.

The algorithm of SemTag, a large-scale annotation system (Dill et al., 2003), combines the results of lexical analysis and the hierarchical relationships between the concepts in the ontology. It uses the TAP taxonomy (Guha and McCool, 2003), which contains more than 72,000 concepts, to annotate about 260 million web documents. After document tokenisation, SemTag tries to map tokens to the taxonomy concepts. Every token is saved with a token signature – ten words before and after it in the text. To distinguish between possible candidate concepts for a certain token, a taxonomy-based disambiguation algorithm is used.

Armadillo (Ciravegna et al., 2004) is another large-scale annotation system. Unlike SemTag, it can use an imported ontology as a source for annotation. The annotation process is based on automatic pattern discovery and information integration. It starts with a set of documents and a predefined lexicon for annotation. Once Armadillo finds a certain pattern (a match between a lexicon element and a content fragment), it begins to look for its confirmation beyond the initial set of documents. If a proper lexicon extension is found, Armadillo starts looking for appropriate patterns for it and then for information confirming these patterns. The information from different sources is integrated, the lexicon is expanded, and the found annotations are stored in the form of RDF triples. The process continues until the algorithm reaches a stable point.

3.4.2 Ontology learning

Ontology learning aims at developing methods and tools for automatic and semi-automatic acquisition of ontologies from structured, semi-structured and unstructured sources. This includes learning of new ontologies from scratch or based on some preformatted sources, as well as enrichment or adaptation of existing ontologies by extending or modifying their structures. Shamsfard and Barforoush (2003) identified several important dimensions for comparing ontology-learning approaches: input data, elements being learned, learning algorithm, and ontology output.

Input data

The starting points for different ontology learning algorithms vary from unstructured corpus of plain-text documents to data with well-defined internal semantic structure (such as existing ontologies, database schemas, and lexical nets such as WordNet). The majority of recent systems attempt to exploit various semi-structured corpora of web resources (HTML, XML and DTD documents, online dictionaries, encyclopedias and thesauri, etc).

Gomez-Perez and Manzano-Macho (2005) provides an extensive survey of ontology learning approaches relying on mining semantic information from unstructured texts. All such approaches are based on advanced NLP techniques complemented with statistical and machine learning algorithms. Many of them employ thesauri information, most often provided by WordNet. Often, text-based algorithms are used as components in a larger environment combining outputs from several evidence providers to learn a resulting ontology. For example, Text-To-Onto (Maedche and Staab, 2001) integrates information from texts, dictionaries, online databases and legacy ontologies.

Online dictionaries and encyclopedias, such as Wikipedia, consist of well-organised corpora of thematic pages interlinked together. A link between two pages usually exists when there is a semantic link between the corresponding concepts. The pages themselves have consistent formatting, which facilitates concept extraction and sometimes identification of hierarchical relations. Several recent systems try to utilise such well-formatted and well-written collections of documents for the purpose of knowledge acquisition. For example, Ruiz-Casado et al., (2005) use patterns from Wikipedia pages to enrich WordNet with new relations. TM4L, an environment for creation and use of topic-focused, ontology-driven learning repositories, supports the authors in building lightweight domain ontologies by automatic extraction of concepts and relationships from the Wikipedia and Wikibooks websites (Dicheva and Dichev, 2007). TM4L uses Wikipedia as a source for proposing concepts relevant to a particular subject and as a source of ‘standard’ concept names.

Elements being learned

Concepts are the basic ontology building blocks. Thus most of the ontology learning systems start the learning process with identification of domain concepts by extraction of appropriate terms from the input corpus. Concepts can also be created during the refinement phase from other concepts. Consequently, two principal ways exist for concept learning (Shamsfard and Barforoush, 2003): terminological concept acquisition and semantic concept acquisition. The first method includes linguistic analysis of the initial set of documents, identification of key noun phrases, word sense disambiguation, and synonym processing. The second method is applied during the ontology refinement stage and includes analysis of the ontology structure (relations between concepts) and concept evaluation (according to their attributes). As a result, some concepts can be merged or broken into smaller concepts.

Generally, relations are harder to learn than concepts, especially from unstructured sources. In such cases, the only available information comes from the morphological analysis of sentences containing identified concepts. If the documents in the initial corpus are connected with
hyperlinks and the content of the documents has internal structure (as in Wikipedia), these structural regularities could be used to connect together concepts found in the linked pieces of the corpus. Different approaches are used for learning taxonomic relations, non-taxonomic conceptual relations and concept attributes.

Axioms or rules are very hard to learn automatically. Few systems are capable of extracting some. An example is Hasti (Shamsfard and Barforoush, 2004), which can learn a limited set of axioms in restricted situations.

Learning algorithm

There are currently no existing approaches for fully automatic learning of heavyweight ontologies from unstructured texts. The automatic algorithms are either developed for enrichment of existing ontologies with new concepts and/or relations (Faatz and Steinmetz, 2002) or for learning light-weight ontologies from well-structured corpus of documents (Apted and Kay, 2004). Most often, ontology learning approaches require human intervention at least on the refinement stage to validate the results of the automatic learning process (Maedche and Staab, 2001).

The algorithms for ontology learning try to find effective combination of techniques from other fields, including computational linguistics, information retrieval, machine learning, and database management. As a result, the algorithms differ along many dimensions (Shamsfard and Barforoush, 2003):

- the degree of linguistic pre-processing (deep NL understanding vs. shallow text processing)
- the degree of automation (supervised/unsupervised/ cooperative)
- the degree of interactivity (online/offline)
- learning approach (statistical vs. symbolic, logical, linguistic-based, pattern matching, template-driven or hybrid methods)
- learning task (classification, clustering, rule-learning, concept formation, ontology population).

Systems supporting the full process of ontology learning usually combine several algorithms to perform a number of intermediate tasks and deal with multiple sources of evidence.

Output ontology

Not all learning algorithms result in the creation of a complete ontology. Some generate intermediate knowledge structures and can be used as components in a bigger ontology-learning environment. For example, the Cameleon system (Aussenac-Gilles et al., 2000) finds the so-called lexico-syntactic patterns, which are good indicators for the presence of a conceptual relation; the final decision about adding a new relation is made by a human expert. WOLFIE (Thompson and Mooney, 1999) learns semantic lexicon from NL texts; it handles polysemy and synonymy and produces a set of terms and representations of their meaning.

Text-To-Onto (Maedche and Staab, 2001), Hasti (Shamsfard and Barforoush, 2004) and MECUREO (Apted and Kay, 2004) are examples of systems learning complete ontologies. However, the resulting ontologies differ very much in their coverage, usage or purpose, content type, structure and topology, and representation language.

4 Ontologies meet user modelling

The benefits of using ontologies for user modelling and adaptation have been recognised by many researchers. Mizoguchi and Bourdeau (2000), in their seminal work listed ten problems of AI research in education and proposed how more formal system design principles relying on ontological engineering could help to overcome those problems. Based on the analysis of the existing research, the authors of Dicheva et al. (2005) developed an ontology of ontological technologies for e-learning and implemented a web portal driven by this ontology, which facilitates the description and discovery of online resources in this area (such as papers, workshops, research groups etc). Winter et al. (2005) analysed the ways ontologies can help student modelling. Among the advantages of ontology-based student models they named

“formal semantics, easy reuse, easy portability, availability of effective design tools, and automatic serialisation into a format compatible with popular logical inference engines.”

A number of dedicated workshops have been organised by several communities (for references see O4E, 2008) and several peer-reviewed journals have published special issues on application of ontological technologies for user modelling and adaptation.

In this section, we present an analysis of this rapidly growing field of research. When possible, we follow the order and structure of the two previous sections. Every subsection tries to demonstrate how a certain user modelling technology benefits of employing particular ontological technologies. Figure 6 shows the structure of the section, the relations between its parts and the corresponding technologies from the fields of user modelling and ontologies. The relations signify only the most important dependencies. (For “Ontology representation technologies” that have influence on every section the links are omitted.)
4.1 Ontologies for user model representation

Ontologies are in the first place a KR technology. Thus, almost any user-adaptive system employing an ontology for user modelling, on some level uses it for representational purposes. Even when the major goal for utilising ontologies is to the benefit of one of the derivative ontological technologies (e.g., using ontology mapping for user model mediation or applying ontology-based learning for automatic construction of user model), the primary design decision of the system developer would be to represent some part of system’s knowledge as an ontology. Therefore, several projects described in this section will be mentioned again, when we talk about ontologies for user model elicitation (Section 4.2) and ontology-based user model interoperability (Section 4.3).

The diversity of ontological approaches for representation of user models springs from both the vast pool of research accumulated in the past on user model representation and the opportunities, ontologies provide for the new generation of adaptive systems. However, we can identify two principle directions summarising the whole spectrum of approaches. First, an adaptive system can employ an ontology for modelling the structure of its domain and then use elements of that ontology to represent atomic user characteristics. The second direction is to structure a complex user profile that models several dimensions of user’s state as an ontology. The first approach inherits its main ideas from the overlay user modelling (Section 2.2.1), while the second one relates to the user modelling dimensions (Section 2.1) and has roots in stereotype user modelling (Section 2.2.3). A small set of research concentrates on facilitating keyword-based user modelling with the use of lexical ontologies such as WordNet.

4.1.1 Ontology-based overlay user modelling

The most conventional implementation of ontology-based user modelling is the overlay user modelling that relies on a domain model represented as an ontology. A simplified view on an ontology is a network of concepts. It provides a natural basis for modelling a domain of discourse following multiple examples in the area of AI and Intelligent Systems. Over the years several network-based KR formalisms have been explored, including Semantic Nets (Woods, 1975), Concept Maps (Novak and Gowin, 1984) and Bayesian Belief Networks (Pearl, 1988). While similar to the
mentioned modelling frameworks in the idea of network-based domain representation, ontologies differ in providing a basis for implementation of sharable and reusable models with rich built-in inference capabilities.

Representation of a domain model as an ontology and modelling a user’s knowledge, interests, needs, or goals as a weighted overlay on the top of it enables the usage of standard representation formats and publicly available inference engines, as well as an access to a vast pool of technologies for ontology mapping, querying, learning, etc. Such straightforward overlay user modelling based on ontologies is implemented, for example, in the misearch system (Speretta and Gauch, 2005a, 2005b; Bouzeghoub et al., 2003). misearch uses the Open Directory Project (ODP) (Netscape, 1998) hierarchy as a reference model to represent interests of a user querying information on the web and to re-rank the search results according to the user’s interest model.

Regional Browsing Agent (RBA) (Chaffee and Gauch, 2000) developed as part of the OBIWAN project (Gauch et al., 2003) supports personalised web-browsing based on users’ interest profiles. As an underlying ontology RBA uses the Lycos hierarchy (Lycos, 1999), which contains 5863 concepts and has a tree depth of four. The concept hierarchy is used to navigate the user to the most interesting websites. The number of stars annotating an item in the hierarchy visualises the system’s opinion about the user’s interest in the corresponding category.

Employing principles of the formal ontological theory, the domain ontologies can also benefit of such properties as intentionality and explicitness, which allows building unbiased and logically complete domain models (Guarino, 1998).

Another possible extension of the overlay user modelling based on a domain ontology is to model user characteristics by employing not only the concepts of the ontology but also its relations and axioms. It could be the knowledge of the fact that a human being ‘is-a’ mammal (relation) or that every person “has two and only two” biological parents (axiom); it can be an interest in anything, which is a ‘part-of’ a particular laptop model, etc. We are not aware of any system employing such modelling techniques. A somewhat related idea is proposed in Sicilia et al. (2004). The authors introduce a concept of learning link for modelling relationships between learning objects and propose an ontology of such links. Concepts of this ontology are used to annotate the hyperlinks in a hypermedia tutorial. The designed prototype system adaptively navigates users taking into account not only the concept knowledge but also the history a user has had for the particular type of links.

The overlay user model can also consist of a subset of concepts from the domain ontology. This is typical for recommender systems representing user interests. The presence/absence of a concept in the user model signifies the presence/absence of interest in the corresponding concept. The concepts can also have associated values (weights, verbal labels), characterising the magnitude of the interest. For example, the museum guide ec(h)o represents user interests as a set of concepts from the ontology of natural history (Hatala and Wakkary, 2005; Hatala et al., 2005). The ontology is taken from the corresponding part of the Dewey Decimal Classification (Dewey, 1876). The strength of an interest is represented by a weight calculated based on the user activity (movements around the exhibition, stops in front of a certain piece, playing a particular audio fragment). Based on the model of interests, ec(h)o recommends to the user the next best audio fragment.

A similar approach is implemented in MyPlanet (Kalfoglou et al., 2001) – a recommender systems for scientific information. The information is disseminated in a form of e-Stories (news items). Users access e-Stories in personalised manner based on their preferences. The preferences are taken from a set of seven ontologies representing research areas, research themes, organisations, projects, technologies application domains, and people. There is no weight associated with a preference; it is a binary value. MyPlanet intensively uses ontological inference to propagate user preference through an ontology and across different ontologies. For example, if the user is interested in a particular genetic algorithm (which is a subclass of the Genetic Algorithm concept, which in turn is a member of the Research Area category), and there is an available news item on another genetic algorithm, this story will be included in the system recommendation. Stories describing Projects or Persons working on genetic algorithms will be also retrieved.

### 4.1.2 Personal ontology views

Many projects using ontologies for modelling characteristics of an individual user go beyond the classic overlay approach. User models in such systems are not represented as simple weighted masks over the domain model, but as individual networks of concepts reflecting personal conceptual structures in the domain of discourse. For models of this kind, Huhns and Stephens (1999) introduced the term ‘personal ontologies’ to designate a computation model reflecting an individual perspective on the world. The term has been criticised later in Kalfoglou et al. (2001) for its contradiction to the definition of ontology as a model representing a shared view on the domain. Accepting the use of the concept itself, however, they suggested another name – “personal ontology view”. The latter term conveys the meaning of the concept more precisely, as the ontology-based user model is not an ontology itself, but its individualised permutation. We are going to follow Kalfoglou et al. and call these models Personal Ontology Views (POV). From a cognitive perspective, POVs are consistent with the constructivist paradigm and cognitive models like Concept Maps (Novak and Gowin, 1984), as they represent domain conceptualisation from the viewpoint of an individual. The POV implementations are different in different systems. We distinguish two basic variations: POV represented as a subnet of a domain
Ontological technologies for user modelling

Ontology and POV as a unique conceptualisation, whose structure can deviate from the underlying domain ontology. The following two subsections analyse these variants in detail.

**POV as a sub-graph of the domain ontology**

The implementation of a user model as a sub-graph of the original domain ontology can be considered as an extended version of the model containing only a subset of domain concepts. In this case, not only concepts but also relations and constraints can be included in the user model. If the model contains only a subset of concepts, the inference of possible semantic connections is somewhat limited, since the actual inference happens in the domain ontology, when the user model can provide only the activation nodes. The sub-graph POV allows a more complete representation of the user model. The inference can be implemented over the POV itself that prunes the irrelevant parts of the domain ontology and contains only the relevant ones.

This approach is realised in two adaptive recommender systems: Quickstep and Foxtrot (Middleton et al., 2004). Both systems recommend research publications. The ontology of research topics used by Quickstep is based on a part of ODP hierarchy (Netscape, 1998). The interest value is computed using four sources of information:

- user explicit feedback (research topic rated as interesting or not interesting)
- user implicit feedback (reading a paper, following system recommendation)
- interest spread over ontology (a parent concept receives 50% of its children’s interests)
- time decay (the importance of a user activity for the interest value is in inverse proportion with the number of days passed since the moment of the activity).

The experimental evaluation of the Quickstep system showed 7–15% greater acceptance rate of recommendations utilising ontology-based user modelling over a traditional overlay user modelling. Foxtrot is a modification of the Quickstep system. It uses a more developed ontology of research topics taken from the CORA project (McCallum et al., 2000), which allows a more precise paper categorisation and more advanced inference.

**POV as individual conceptualisation**

The second version of POV is a completely individual knowledge model that can differ from the original domain ontology in its structure and concept names. There are few systems employing such a ‘pure’ implementation of POV.

**OWL-OLM** (Denaux et al., 2003b) is a user modelling component used for elicitation of student knowledge through interactive dialogues about the subject domain. Based on her or his utterances a dialog agent builds a POV representing an individual student’s conceptualisation. It may contain relationships not existing in the domain ontology, but reflecting student understanding of this part of the domain. By comparing the domain ontology and the elicited student conceptualisation, OWL-OLM identifies the faulty relations, localises the problem, and infers the pattern responsible for the error in the conceptualisation. After that, the dialog agent chooses the next utterance, which could be one of five types:

- give answer to the user’s question
- inform the user about an ignorance
- agree with the user’s utterance
- disagree with the user’s utterance
- ask the user a question.

The user modelling in OWL-OLM is divided into a long-term model and a short-term ‘conceptual state’. The ‘conceptual state’ is built during a session; after the session ends, OWL-OLM uses the ‘conceptual state’ to update the long-term model. OWL-OLM recognises 17 types of mismatches, such as “The same concept in the domain ontology is not associated with this term”, or “The domain ontology does not suggest that two concepts are related by this property”. The current version of the system works in the domain of simple Linux commands. However, the choice of OWL as a representation model and the system architecture make OWL-OLM truly domain-independent.

The authors of the mentioned OBIWAN project (Gauch et al., 2003) have experimented with several approaches to ontology-based modelling of interests of users searching information on the web. Their implementation relying on a simple overlay approach was already discussed in Section 4.1.1. Another system developed in the same project supports personalised web browsing based on user models edited by the users themselves. The system asks the user to specify a small hierarchy of topics representing their general search interests. The user is limited neither in the choice of text labels for the topics, nor in the structure of POV. The system then uses POVs to navigate users to pages of their interest. To personalise the web content, OBIWAN associates the pages being browsed with the topics of interest from the POV. As an intermediary it uses a reference ontology based on the subject hierarchies from Yahoo! (Kimmel, 1996), Magellan (McKinley-Group, 1997), Lycos (Lycos, 1999) and ODP (Netscape, 1998) and multiple web resources assigned to the topics from these hierarchies. OBIWAN uses an ontology mapping technique to build the connection between the POV representing the model of user interests and the reference ontology. As a result, all web pages from the reference ontology are linked to topics from POV and the user can browse them in a personalised way.

**4.1.3 Ontologies of user profiles**

Sharability and reusability of user models require not only standardised domain representation but also common vocabularies for describing the user model itself. Therefore, the design of ontologies describing sharable user profiles has become an important research direction related to the application of ontologies in user modelling. The term
A standardised user profile will make possible the implementation of interoperable adaptive systems sharing modelling information. Ontology-based user profiling is especially important for systems reasoning across multiple profiles (social adaptive systems) or for systems that can benefit from complex inference on multiple ontologies representing different knowledge (e.g., adaptive pervasive systems using ontologies to model domain and context knowledge).

Ontologies for learner profiles

The main motivation of the FOAF project is to provide a metadata schema for implementation of social networks on the SW.

Several attempts have been made to integrate previously developed models into a single synergetic representation of a user. Dolog and Nejdl designed an architecture for ontology-aware AES on the SW. One of the ontologies in this architecture is the learner profile ontology, developed as a combination of IEEE PAPI and IMS LIP (Dolog and Nejdl, 2003). This ontology was extended with new concepts for the implementation of the TANGRAM learning system (Jovanovic et al., 2006). Onnas et al. (2006) analysed the applicability of several existing approaches (including the already mentioned IEEE PAPI, IMS LIP, Dolog and Nejdl’s student profile ontology and FOAF) to the development of Learner Profile in Social Software. They decided to use the superset of these models taking FOAF as a basis.

Ontologies for generic user profiles

The aforementioned projects have mainly focused on representing information about learners. However, a number of research teams try to address the more general problem of developing a unified way of modelling a general user. An important difference here is the undefined type of the user and the context of system usage. The target user of AES is a student acting in an educational setting with a main goal to study. The modelling of a generic user or user acting in arbitrary context should take into account the context and the role of the user.

Niederée et al. (2004) proposed the Multi-Dimensional Unified User Context Model (UUCM) and the creation of a user-modelling server for cross-system personalisation. To provide the maximum flexibility of modelling, UUCM supports several modelling dimensions for modelling different aspects of users’ state and activity: user context, task, cognitive pattern, relationship and environment. The structure of UUCM consists of two levels: the outer level is a meta-model represented as an upper-level ontology describing the different dimensions and relationships among them. On the inner level, every dimension is represented using its own ontology. Such composite architecture allows adaptive systems communicating with UUCM to concentrate only on the dimension they need. It also eases the expansion of the model with new dimensions.

Heckmann developed UbisWorld, an infrastructure for cross-system ubiquitous personalisation on the basis of a set of shared ontologies (Heckmann, 2006). The adaptive systems in UbisWorld communicate with each other by means of messages, carrying user modelling information expressed in UserML, an XML dialect described in Heckmann and Krueger (2003). UbisWorld uses a large set of ontologies describing various aspects of the reality that can be employed for ubiquitous computing. Figure 7 presents these ontologies along with classes of objects modelled by them. One of the components modelling the current user’s situation is General User Model Ontology (GUMO) (Heckmann et al., 2005).
GUMO defines several hundred of concepts describing various user characteristics from traditional ones, like demography or knowledge to such, rather exotic, as facial expression (see Figure 8). Several applications are currently using this platform, including MobileMuseumGuide (Kruppa et al., 2005) and Ubidoo (Stahl et al., 2007). The Ubidoo system implements the functionality of a personal organiser (Calendar + To-do list). The organiser adapts its alarms and directions based on the changes in the user’s location. If a user sets an alarm from one place, but at the time of notification, she or he is in another location, the time necessary to complete a task is updated according to the distance to the target place (e.g., airport). The target place necessary to complete a certain task can be updated as well (e.g., if the task is to shop for groceries, Ubidoo can recommend different grocery stores depending on the current location).

**Figure 7** UbisWorld ontologies (see online version for colours)

![UbisWorld ontologies](source.png)


**Figure 8** Top-level concepts in GUMO (see online version for colours)

![Top-level concepts in GUMO](source.png)


### 4.1.4 Ontologies for stereotype user modelling

A number of papers combine the use of ontologies with user modelling based on stereotypes. The main premise of stereotype user modelling is the existence of typical user categories with generalisable characteristics (see Section 2.2.3). The actual users coming to the system are classified into one or several such categories, and their user models inherit some parts from the corresponding stereotypical profiles. The use of ontologies in this process can be three-fold. First, the domain ontology can be used to populate stereotype profiles; second, a single stereotype profile itself can be implemented as an ontology; and third, the ontologies can help organise the stereotype structure to improve inter-stereotype reasoning.

Several systems applying ontologies for stereotype user modelling rely on a domain ontology to populate the characteristics of predefined stereotype profiles. The models of individual users can be represented as overlays based on the same domain ontology. When the individual user model matches the preconditions of a certain prototype, this prototype’s profile updates the user model. Ardissono et al. (2003, 2004) describe a user-adaptive TV program guide Personal Program Guide (PPG). PPG stores stereotypical information about TV viewer preferences for such categories of users as, for example, housewife. The TV preferences themselves are modelled by a ‘general ontology’, which includes hierarchy of TV programs from broad like ‘Serial’ to more specific like ‘Soap Opera’, “Sci-Fi Serial”, etc. The use of an ontology allows PPG to structure the domain knowledge and effectively map different TV program characterisations to the ontological categories. Gawinecki et al. (2005) presents an ontology-based travel support system. The system utilises two ontologies for describing such travel-related domains as hotels and restaurants. User modelling is consistent with the traditional stereotype approach – particular preferences of a user are inherited from stereotype profiles. The stereotype profiles themselves are populated based on the ontologies. This project signifies another benefit of using ontologies – both ontologies have not been developed by the authors but reused from relevant projects.

Recent advancements in the development of ontologies for user profiles open opportunities for formalisations of single stereotype models as separate ontologies. An implementation of a stereotype as an ontology makes it a sharable and reusable unit and can lead to the creation of libraries of such modularised stereotypes. Both projects mentioned above model the stereotypes as small ontologies, however they do not base them on any existing initiative for user profile ontologies. Gawinecki et al. (2005) design their stereotype ontologies using RDF.

Some classic systems, like Grundy (Rich, 1979), exploited hierarchies of stereotypes long ago. Such structures facilitate stereotype management and make the stereotype transition more effective. The more specific stereotypes inherit characteristics from the more general ones, hence a profile of a user switching, for example, from a religious person to a Christian does not undergo unnecessary updates. Systems implementing an ontology of stereotypes get access to the pool of ontological technologies for stereotype processing. Reasoning across stereotypes as in the example described above can be performed with the use of publicly available ontological inference engines. Different systems employing local ontologies of stereotypes can rely on ontology mapping techniques to facilitate mediation of user models across the systems. If a system decides to open the user profile to the user, or even allow users to specify what stereotype they think they belong to, ontology visualisation techniques
could be employed for implementing the corresponding part of the system interface. Nebel et al. (2003) propose the implementation of a database of reusable user ontologies as a source for interoperable user information.

4.1.5 User modelling based on lexical ontologies

Thesauri and lexical ontologies, such as WordNet, allow systems implementing traditional keyword-based user modelling (see Section 2.2.2) to automatically enhance user models with shallow semantics based on lexical relations between words. This also provides a ready solution for such NLP problems as polysemy (several meanings of the same word) and synonymy (several words with the same meaning). WordNet can considerably improve the quality of keyword-based modelling of content, as its synsets ensure unambiguous interpretation of NL terms. Systems that implement keyword network modelling can utilise relations between WordNet synsets, which allow them to build more meaningful term networks than those based on the term collocation.

One of the first adaptive systems exploiting WordNet for the representation of user models is SiteIF (Stefani and Strappavara, 1998; Strappavara et al., 2000). SiteIF is an adaptive news recommender system “watching over the user’s shoulder” while she or he browses the news items. It uses the browsing history to predict users’ interests and recommend relevant news. Unlike many other news recommender systems that model news content by plain keywords, SiteIF employs a set of techniques for word sense disambiguation. For these purposes it uses MultiWordNet, which is an Italian version of WordNet with some extensions. Synsets in MultiWordNet are annotated with semantic labels. A label represents one of 250 fairly large categories (e.g., Architecture, Medicine, etc.). Based on such an association not only the content of news items can be modelled in terms of word senses, but they can be also classified as belonging to one of the categories. A user’s interests are modelled as a semantic net, where the nodes represent WordNet synsets and the arcs – co-occurrences of two synsets with arcs’ weights modelling co-occurrence frequencies.

Semeraro et al. (2007) exploit a similar approach. They introduce the JIGSAW algorithm that uses WordNet as a tool for word sense disambiguation to move from flat keyword-based user profiles to ‘sense-based’ user profiles. JIGSAW adopts several previously developed algorithms to appropriately disambiguate nouns, adjectives, and adverbs. Its distinctive feature is the utilisation of the hierarchical structure of WordNet. Consequently, not only it takes into account the presence of relevant keywords in the same synset, which helps to deal with problems of polysemy and synonymy, but also tries to benefit of such relations between synsets as antonymy (‘opposite’), hyponymy/hypernymy (‘is-a’), and meronymy (‘part-of’). The evaluation of JIGSAW has demonstrated that the accuracy of adaptive recommendation driven by ‘sense-based’ user modelling is considerably higher than that based on traditional keyword user models. Authors report increase in both precision (8%) and recall (10%).

The same research team also applied WordNet-based user modelling for the implementation of a hybrid recommender system (Lops et al., 2007). Hybrid recommender systems (Burke, 2007) combine several recommendation approaches in attempt to compensate weaknesses of one method by the strengths of another. For example, the quality of recommendation based on collaborative filtering suffers from the so-known ‘cold start’ problem. It is hard to find like-minded users for a new user, who just has joined the system and have not yet generated enough evidence for the system to make a valid prediction. Content-based recommendation can help to remedy this problem, as it is based on the content model of information items and can start working immediately after a user has made his first move (e.g., has rated the first item). However, the quality of content-based recommendation strongly depends on the quality of item models. Lops et al. (2007) apply WordNet synset to model the content of movie descriptions. The descriptions contain several slots: title, genre, cast, short summary, etc. Collaborative filtering is applied to compute similarity between users based on the users’ ‘sense-based’ profiles. The evaluation has shown statistically significant improvement in the predication of like-minded users that this approach brings comparing to the traditional collaborative filtering.

WordNet, has proven to be a useful tool for word sense disambiguation and shallow semantics disambiguation in many applications (Rosenzweig et al., 1999). Since its first release, its authors have been improving its structure and advancing its semantic capabilities (see, for example, Harabagiu et al., 1999). Starting from version 2.1, WordNet allows to distinguish such ontological categories as classes and instances (Miller and Hristea, 2006), i.e., it is capable to differentiate between relations ‘is-a’ and ‘subClassOf’. This feature allows systems using WordNet to move one step closer towards an automatic generation of semantic content profiles from unstructured text.

4.2 Ontologies for user model elicitation

This section describes projects that can be roughly divided into two categories: projects using ontology-based knowledge acquisition techniques for unobtrusive user modelling and projects applying ontology visualisation for open, editable, and interactive user models.

Unobtrusive user modelling utilises a set of technologies from such fields as machine learning, data mining, and information storage and retrieval. As a result, adaptive applications trying to elicit user information implicitly, in unobtrusive manner, often face challenges arising due to the conflicts between the computational demands of machine learning and data mining algorithms, on one side, and the interactive and dynamic nature of the user modelling task, on the other. These challenges have been nicely summarised by Webb et al. (2001) (Section 2.3.2). For some applications, the problem can be aggravated by
Ontological technologies for user modelling

additional requirements of the adaptation task, such as explanation ability and semantic interpretability of user modelling assumptions. Recent research efforts on application of ontological technologies for unobtrusive user modelling demonstrate the potential to partially overcome some of those difficulties.

Systems allowing users to view, explore, and modify their user models can benefit from using ontology-based visualisation. An ontology can help to structure and externalise the conceptual knowledge in the domain. Hence, the utilisation of an ontology as a cognitive component can help an open user modelling system to communicate domain knowledge to the user in a natural way.

In this section we discuss using ontological technologies for population of user models with individual information and for mining user models with individual structures for adaptation purposes. We also overview works on using ontology visualisation to open user models and help users to explore and modify their models.

4.2.1 Ontology-based population of user model

One of the major challenges of content-based adaptation is the efficient modelling of the content in terms of domain elements. Two main approaches developed over the years address this task from different angles. The classic KR methodology requires building an external model of the domain semantics and manual indexing of content items with elements of this model. The resulting representation allows associating users’ actions (solving a problem, accessing a document, etc.) with knowledge underlying the accessed information items and implementing effective adaptation strategies. However, this authoring approach is time-consuming and expensive, and not applicable for many applications operating in the setting of AW.

Another methodology that originated in the field of information retrieval and data mining does not require the creation of sophisticated models, nor does it need the intervention of domain experts to provide manual content indexing. Instead, it relies on the automatic extraction of keyword vectors from the documents’ content. This approach has been successfully applied for a number of tasks requesting automatic modelling and effective retrieval of textual information (e.g., web search); however, it has its own problems as well. The keyword-based modelling is not capable to deliver the elements of domain semantics. As a result, the adaptation strategies that require deterministic reasoning do not work. The alternative machine learning approaches impose additional requirements and are not always applicable in adaptive applications.

Recently developed ontology-based annotation techniques look promising for building a bridge between semantically rich concept-based models and automatically generated keyword-based models.

Three different communities contribute to this field of research forming two main perspectives on the problem. Ontologies and traditional adaptation technologies share the heritage of AI and classic KR. From their point of view, the ontology-based annotation helps to augment the traditional semantic indexing with automatic capabilities. For the adaptation community, it opens an opportunity to solve the classic problem of open-corpus personalisation. The IR and DM community appreciates ontology-based content annotation for the generation of semantically rich models. The main component of user models developed in this field is traditionally a flat list of user transactions. Therefore, the projects originated in this community often refer to the problem as a semantic enrichment of usage logs. Ultimately, both viewpoints represent the same functionality – automatic generation of semantically rich content and user models. The difference is in stressing ‘automation’ vs. ‘semantics’.

Below we first discuss projects addressing the problem of open-corpus personalisation with the help of ontology-based automatic content annotation and then review works moving from the traditional keyword-based user modelling to the concept-based approach through semantic enrichment of usage logs.

4.2.2 Ontology-based open-corpus personalisation

The observed interactions of a user with an adaptive system’s content may result in a collection of detailed transactional logs. The challenge for the adaptive system is to interpret this data and utilise it for adaptive tailoring of future user interactions. Traditional content-based adaptation relies in this process on composite domain models and associations between content items (e.g., tutorial pages, problems, etc.) and domain elements (concepts). It allows adaptive systems to reason in terms of domain semantics instead of content, and to generalise user characteristics based on their history of interaction with the system. For the closed-corpus applications operating on restricted, relatively small sets of content items, the association of such items with domain elements can be done manually with the help of advanced authoring tools. However, AW user modelling systems usually operate in open-corpus settings; they have to deal with virtually unlimited or dynamic sets of documents and cannot benefit of manual authoring. Consequently, when the corpus size goes beyond a manually manageable threshold (as in adaptive web search) and/or the content refreshes on a regular basis (as in adaptive news recommendation), content-based adaptation requires automatic support for semantic indexing. A promising direction to remedy this problem is to apply ontology-based annotation techniques for automatic or semi-automatic indexing of the content to be adapted.

Open Adaptive Hypermedia System (OAHS) (Henzel and Nejdl, 2002) is one of the first attempts to apply ontologies for the problem of modelling and adaptation in an open-corpus environment. The authors do not specify however how OAHS supports semantic mark-up of unrestricted content. They discuss mechanisms for estimating users’ knowledge and implementing adaptation strategies in the conditions of an open-corpus document space.
A number of projects adopting this approach to personalisation come from the field of ontological research. One of them is Conceptual Open Hypermedia Service (COHSE), a joint project between Sun Microsystems and University of Manchester (Bechhofer et al., 2003; Yesilada et al., 2007). COHSE is based on the original idea of distributed link services (Carr et al., 1995) working as intermediaries between web clients and web servers and augmenting web documents with dynamic links. Architecturally COHSE works as a proxy. It intercepts a client’s HTTP request, retrieves the original HTML document, enriches its content with new links and delivers the modified document to the user’s browser. The extra links added by COHSE connect key pieces of text in the original document with relevant HTML context elsewhere. This knowledge-driven linkage is performed on the basis of an ontology, serving as a source of consensual domain semantics. The COHSE components apply ontology-based annotation techniques to associate automatically pieces of documents with ontology concepts. When a document is requested, its annotations act as links to concepts and thus to other documents annotated with the same (or relevant) concepts. Currently COHSE does not support user modelling and user-based adaptation of the content. The proposed links are based ultimately on the hidden domain semantics and do not take into account individual browsing history or other sources of user-related information. However, recent publications on COHSE report interest in this research direction and list it in the plan for future development.

Similar approach is implemented in the project Magpie ( Domingue et al., 2004; Dzbor and Motta, 2007). Unlike COHSE, Magpie works on the client side as a browser plug-in. It analyses the content of the HTML document being browsed on-the-fly and automatically annotates it based on a set of categories from an ontology. The resulting semantic mark-up connects document terms to ontology-based information and navigates a user to content describing these terms. For example, if Magpie recognises that a particular phrase is a title of a project, it populates a menu with links to project’s details, research area, publications, members, etc.

By combing the capabilities of systems like COHSE and Magpie with the architectural solutions provided by OAHS, one can obtain a complete functionality necessary for open-corpus personalisation. Several systems attempted to solve this problem for the task of adaptive recommendation.

The systems Quickstep (Middleton et al., 2001) and Fox Trot (Middleton et al., 2003) mentioned in Section 4.1.2 apply a similar set of technologies for implementation of adaptive recommendation of research papers in open-corpus settings. Newly found papers are automatically classified as belonging to one of the topics from the research topic ontology. If the topic matches some of the interests in a user profile, the paper is recommended to the user. The profile is updated based on the user’s activities (browsing papers, downloading papers, rating papers, following recommendations).

Another example of fully implemented user modelling and adaptive recommendation for unrestricted content is provided by the system MyPlanet (Kalfoglou et al., 2001), created in the Knowledge Media Institute (KMi) of Open University (UK) by the same research team who developed Magpie. MyPlanet provides personalised access to KMi corporate news e-stories based on user preferences. On the meta-level user preferences belong to one of the following ontologies: research areas that are investigated in KMi; research themes that are investigated in KMi; organisations that KMi collaborates with; projects in KMi; technologies used in KMi; application domains that are investigated in KMi; people - members of the KMi lab. The e-story submission is done by e-mail. Once submitted, stories are automatically annotated by ontology concepts using one of 40 predefined event templates. For example, for the event “visiting-a-place-or-people”, the template looks as follows: [_, X, _visited, Y, Y, from, Z, _]. Here X is an entity capable of visiting, Y is a place being visited, and Z is for dates of the visit. When MyPlanet recognises a story matching this template, it extracts participating entities, associates them with ontology concepts and annotates the story accordingly. When selecting e-stories to present to the user, MyPlanet tosses queries to the ontology and retrieves story items matching the user profile. If the system finds available stories annotated with concepts, which are neighbours of some of the concepts defining user’s interests, such stories will be also retrieved.

### 4.2.3 Semantic enrichment of usage logs

The same problem can be addressed differently, from the viewpoint of traditional data mining and web-based recommender systems. Data mining research usually deals with large sets of documents; consequently, the focus here is not on the size of the annotated corpus or the automatic generation of content models, but on the semantic annotation. Conventional approaches to adaptation in this field do not presume semantic annotation of content items, therefore the papers applying ontology-based annotation for adaptation in this field talk about semantic enrichment of transactional logs or semantically enhanced recommendation.

Oberle et al. (2003) specifies a general algorithm employed by adaptive systems relying on mining of conceptually enriched usage data:

- **Description of raw data**: the description of the transaction, which is generally a user visit or session, as a set or sequence $T$ of URLs $u: T = \{u_i\}$ or $T = [u_i]$
- **Mapping**: the mapping of URLs to objects that are meaningful within the application domain (e.g., concepts from an ontology), with $m(u) = O$ describing the mapping of URL $u$ to a set of objects $O$
• **Mining**: the identification of patterns in the set or sequence of these meaningful objects, i.e., in the transformed transaction $T' = \{U_i, O(u_i)\}$ or $T' = \{O(u_i)\}$.

The difference of this approach from the traditional data mining for user modelling is the utilisation of semantic information in the user modelling and adaptation processes. This enables discovery and recommendation of usage patterns, inferred on the basis of the semantic equivalence of pages. It also allows adjustment to the new and/or dynamic content and changes in the structure of a website. Semantic links do not break (however, preprocessing stage should be repeated).

For example, the SEWeP web recommender system (Magdalini et al., 2004) performs the following routine to issue more meaningful recommendations, based on the content semantics. All web pages are represented as weighted term vectors; the terms are mapped to the concepts content semantics. All web pages are represented as weighted vectors of taxonomy concepts. After that, all documents are combined into semantically coherent clusters based on the obtained vectors; taxonomic relations are used to cluster together documents described by close concepts. Correspondingly, SEWeP augments web-logs of user activities with conceptual information for the visited pages. Association rules mined from the web-logs represent causative relations between concepts. These rules are used for generating recommendations to users who have generated a sequence of transactions corresponding to the left part of a rule.

Similar solution is implemented in Dai and Mobasher (2002). The authors do not concentrate on the details of associating semantic metadata with web pages. The focus is instead on developing a framework for discovery and representation of complex domain-level aggregate profiles based on the usage data and the semantics underlying the web pages. The domain is represented as a heavy-weight ontology with classes and objects combined in a complex network through typed attribute relations. As a result, much richer information becomes available for the inference of semantic user profile.

Zhang et al. (2007) developed Semantic Web Usage Log Preparation Model (SWULPM) that helps recommender systems to integrate usage data and domain semantics represented by an ontology. The evaluation of the recommendation quality based on SWULPM demonstrated about 10% accuracy increase in comparison with recommendations based solely on the usage history taken from shallow sessions.

Utilisation of domain ontologies enables adaptive recommender systems to combine different (somewhat orthogonal) recommendation strategies: the search for possible items to be recommended can be performed both in the previously collected usage history (collaborative filtering approach) and in the rich domain ontology (knowledge based approach) (Mobasher et al., 2004).

### 4.2.4 Ontology-based learning of user model structure

The previous section overviewed systems enriching content with semantic information and using ontologies for deriving generalised user characteristics. In these cases, the underlying ontology defines and restricts the structure of the user model; the modelling system solely populates it with information reflecting the user’s idiosyncrasies. A number of recent research projects attempt to achieve a more challenging goal of mining the structure of the user model itself based on the user’s activity. To facilitate the complete creation of user models from the observed implicit user behaviour, these systems apply a set of techniques related to ontology learning and ontology mapping. User models that are not restricted by a domain ontology have a potential to define fully individualised views on the domain semantics, which can have their own structure and even vocabulary. We have discussed partly such models in Section 4.1.2. Not all approaches described here address the challenge of creation of a complete ontology-based user model; some works only report preliminary progress in this direction.

Zhou et al. (2005) described a system generating personal “web usage ontologies” based on transactional logs. The multi-stage mining process the authors propose utilises log analysis, fuzzy logic, formal concept analysis, and graph models. After a log’s pre-processing (including transaction filtering, session separation, etc.), the so-called web-usage context is calculated for every session – the sessions are annotated with fuzzy categories representing the time of the session and a broad topic browsed by the user during the session. The next two steps – construction and pruning of web usage lattice – result in a graph representing all important navigation states during the session, connected to each other if a transition between corresponding states is possible. The importance of the state is defined on the stage of pruning based on the support and confidence values for the state. Finally, the pruned web lattice is used to generate a “web usage ontology” of the user represented in OWL. As a future research direction, the authors plan to implement an adaptive service utilising the described approach. Unfortunately, the resulting knowledge structure can hardly be named an ontology in the conventional sense, as it does not represent the declarative semantics of a domain. The model rather contains a set of mined navigation patterns and transition rules. Nevertheless, this project demonstrates a potential for extraction of rich models based on the deep web usage mining.

Several projects, originated in the field of user modelling, apply ontology learning techniques to learn the structure of the domain ontology and then use it for adaptation purposes. For example, the ontology learnt by MECUREO (Apted and Kay, 2004) (which we mentioned in Section 3.4.2) has been applied in a number of adaptive systems (for example, for visualisation and navigation (Apted et al., 2003a) and for learning object metadata.
generation (Apted et al., 2004). Zouaq et al. (2007) reports initial progress on the Knowledge Puzzle system that tries to learn domain ontologies from unstructured text. The authors plan to utilise the learnt ontologies for the purpose of adaptive instruction.

The Willow system (Pérez-Marín et al., 2006a, 2006b, 2007) is the only known to us example of learning the conceptual structure of a user model from unstructured text. Willow is an automatic assignment grading system assessing free-text students’ answers. When a teacher creates assignments in Willow, she or he needs to specify to which of the top-level domain categories the assignment belongs. Every assignment also requires at least three sample answers provided by different experts to build a reference answer model and take care of possible rephrasing. When a student submits her or his own answer to a problem, Willow extracts the concepts covered in the answer and matches them against the concepts learnt from the reference model. As a result, it creates an individual conceptual model of a student reflecting the correctly reported knowledge and possible misconceptions. The evaluation of Willow shows that there is a statistically significant correlation between the grades given by the system and the actual grades received by the student in class.

A relevant approach is implemented in the OBIWAN project, previously mentioned in Section 4.1.2. It does not mine the user’s POV from her or his actions, instead, it allows the user to create an individual model of the domain and maintains AW browsing based on this model (Gauch et al., 2003). As already mentioned, the OBIWAN user can specify a small hierarchy of topics representing her or his interests, using any topic names and structuring the model in any way she or he prefers. To create a common ground for various interest models OBIWAN uses a large reference ontology aggregating several available web directory hierarchies. The reference ontology concepts come with web pages attached to them. OBIWAN analyses these pages to extract keyword profiles for the reference ontology concepts. When users define their interest model, the system asks them to provide several sample web pages for every topic in the individual model. OBIWAN uses this data to map concepts from the reference ontology to the concepts in the individual user interest profile. The found mappings allow OBIWAN to model users’ interests and navigate a user based on the newly created topic hierarchy that reflects a completely individualised view on the domain not restricted by a pre-existing knowledge model.

A set of systems previously mentioned in Section 4.1.5 utilises lexical ontologies like WordNet for eliciting sense-based vs. keyword-based user models. Such projects as StielF (Stefani and Strappavara, 1998; Strappavara et al., 2000) and JIGSAW (Semeraro et al., 2007) pre-process keyword-based models by linking synonymous words to the same synsets and disambiguating senses of polysemous words. The synsets in the models can be connected by WordNet relations (antonymy, hyponymy, and meronymy). WordNet extensions, such as MultiWordNet, allow automatic attribution of a synset to a certain top-level domain category. The resulting models represent individualised profiles of users’ characteristics; their structure depends on the history of users’ actions (e.g., in StielF the model of a user’s interests is based on the news items a user has browsed).

### 4.2.5 Ontology-based open, editable and interactive user modelling

In addition to powerful representational capabilities, ontologies can provide a useful way for presenting a domain knowledge to the user in a comprehensible and navigable form. Several adaptive systems have exploited this feature to visualise the domain semantics and directly communicate to the users the current state of their user models. Two dominant ontology visualisation layouts are used by these systems – a class hierarchy and concept maps – however, some other techniques have been tried as well. Some systems use ontologies solely as a navigation aid, others implement ontology-based interfaces to open user models to the users, and finally, a limited set of systems exploit ontology visualisation to elicit user characteristics in an interactive manner.

The OWL-OLM system (previously discussed in Section 4.1.2) provides an example of using ontology visualisation to elicit users’ knowledge of domain concepts and relations between them (Denaux et al., 2005b). OWL-OLM organises the process of student knowledge elicitation in the form of interactive dialog between the student and the Dialog Agent. The Agent analyses the student knowledge model and chooses a concept, not known by the student, or for which the student demonstrated a misconception. The Agent maintains the dialog to assess student knowledge of the entire neighbourhood of the target concept, including the concept attributes and its relations with other concepts in the domain ontology. Figure 9 shows a part of the dialog about the file system in Unix. The Dialog Agent states its utterances in a text form. The student creates response utterances using a graphical editor thus building gradually the target part of the ontology. The system converts student’s actions into OWL statements and verifies them against the domain ontology. If it finds a mismatch between the student model and the domain ontology, the dialog is altered to resolve this mismatch. The dialog ends, when the Dialog Agent fills that it has properly assessed all knowledge relevant to a particular concept, or it can be ended by the student. The important feature of the OWL-OLM approach (besides the interactive graphical ontology-driven dialog-based elicitation of student knowledge) is the employment of ontologies as a learning component. By creating and browsing their own versions of the domain ontology in the form of concept maps, students learn directly the conceptual structure of the domain.

The Verified Concept Mapping (VCM) system (Cimolino et al., 2004) follows a similar approach. Students working with VCM demonstrate their knowledge in the domain by building its concept map. The student interface
of VCM consists of a working space, where students draw their maps, a list of predefined concepts and a list of predefined relations (Figure 10). The teacher authoring the problem should populate these lists. If a student feels that some of the concepts and/or relations are missing, she or he can add their own concept map elements. After the student submits the map, VCM checks its correctness against the etalon map created by the teacher, according to the assessment rules specified also by the teacher. If the student’s map violates any of the rules, the system generates appropriate feedback. The correct passing of certain requirements is also included in the feedback. Application of concept mapping to elicit conceptual structures of users’ domain knowledge is a popular approach (see also the COMPASS system (Gouli et al., 2004)). The difference of VCM is in the utilisation of concept maps for population of an ontological student model, which is verified against the domain ontology.

Figure 9 Ontology-based elicitation and visualisation of user models in OWL-OLM (see online version for colours)

Source: Denaux et al. (2005a)

Figure 10 POV assessment by Verified Concept Mapping

Source: Cimolino et al. (2004)

VCM is in the utilisation of concept maps for population of an ontological student model, which is verified against the domain ontology. COMOV opens student models not for the students to reflect upon them, but for the teacher to monitor student performance. COMOV can present the model of a particular student and the aggregated model of the class. It experiments with four different kinds of visualisation: concept map, table, bar chart (or skillometer) and text summaries (see Figure 11). The node colours represent the conceptual knowledge of the student or the class on the scale from red (no knowledge) to green (maximum knowledge). The knowledge levels of leaf-concepts are aggregated to propagate knowledge for top categories in the ontology. The evaluation of COMOV with real teachers demonstrated positive attitude towards the system. Teachers appreciated the opportunity to observe students’ performance on the concept level. The most popular visualisation layout was concept maps. This finding is consistent with a similar experiment done for the Flexi-OLM system. According to the student evaluation of several layouts for opening conceptual knowledge models, concept maps and the lecture structure have been found most useful.

Figure 11 Student model visualisations in COMOV (see online version for colours)

Source: Pérez-Marín et al. (2007)

The VIUM project (Uther and Kay, 2003; Apted et al., 2003a, 2003b) proposes an interesting solution to the problem of visualisation of large ontology-based user models. When the size of the domain ontology is above several hundred concepts, visualising the user model can become a challenging task, as the capabilities of the screen as well as the perception capabilities of the users are limited. VIUM visualisation (see Figure 12) allows users both to have a snapshot of the entire model and to concentrate on the concept that is currently in focus. The visualisation applet occupies a fixed area on the screen and can present as many concepts as needed. When a user clicks on a concept, VIUM highlights the concept by increasing the label font and reserving some space around it. The related concepts are highlighted in the same way, but with less magnitude. The rest of the concepts, not relevant to the current user focus, are presented in a small font and very close to each other. When a user browses the hierarchy, all concepts are clickable, and once a user clicks on a concept, the focus shifts to it. The relevant positions of concept labels represent the distance between systems.
the concepts in the model; the left indentation shows the level of the concept within the hierarchy.

Essentially, the VIUM visualisation is a smart modification of the traditional class hierarchy, which allows keeping all concepts on the screen and accessing them in one click. The concept labels’ colour represents the user’s interest in the concepts and changes on the scale from red (minimum) to green (maximum). In addition, VIUM allows users to filter out some relation types in the original model and to adjust the colour-switching thresholds and the depth of the relevant concepts highlighted together with the focus.

Figure 12  Visualising conceptual user models in VIUM (see online version for colours)

Some systems exploit different ontology visualisation techniques. For example, QuickStep allows a user to navigate an ontology of topics using a set of pop-up menus (see Figure 13). If the user does not agree with the result of the automatic annotation provided by QuickStep, she or he can manually modify the annotation by choosing a different topic from the menu hierarchy. This will influence the calculation of the topic interests for the users based on their activity with the current paper.

Figure 13  Quickstep allows users to browse the ontology and change the topic associated with the paper (see online version for colours)

The Foxtrt recommender system opens for a user not only the current state of her or his interest model, but the evolution of the user’s interest in a topic. The visualisation is implemented as a time chart (see Figure 14). The user can view several interest profiles at the same time. Foxtrt also gives the user the opportunity to modify the profile chart, which results in the modification of the actual user profile and reinforces the modelling mechanism. The evaluation of this approach showed that direct profile modification helped Foxtrt to improve the accuracy of recommendations during the first week. These results are not surprising, as in the early stage Foxtrt has not accumulated yet enough information about users based on their activity, therefore the direct profile modification provides the Foxtrt’s modelling component with very useful information. Consequently, the modification of the current levels of user interests would require less effort on the user’s side and probably would have the same effect. The usefulness of the modification of user profile’s history and applicability of this approach in specific settings (system tasks, user population, etc.) requires more research.

Figure 14  Use profile visualisation in Foxtrt (see online version for colours)

4.3 Ontology-based user model interoperability

Nowadays, WWW becomes the major infrastructure for information and service delivery, including adaptive services. Many web systems collect data about their users, maintain models of certain user aspects and tailor their behaviour accordingly. The situation, when the same user works with multiple adaptive systems is very common. For example, the same user can receive personalised recommendations from Amazon.com when shopping for books and from Blockbuster when renting movies. The quality of recommendations will highly depend on the amount of information both systems have about the user; and, in many cases, the information collected by one system can be relevant for the judgements made by another. For example, if the Blockbuster recommender was aware that the user bought on Amazon.com all books about Harry Potter it would have the evidence that the user might...
be interested in renting a new Harry Potter DVD, and probably some other DVDs with Fantasy movies. Unfortunately, since the systems cannot get an access to each other’s user model collections, they have to extract user interests on their own. To make cross-system adaptation possible, mediation of user models between the systems is necessary. This task involves several important issues including privacy, security, trust, user identity, etc. However, probably, the central challenge here is how to translate between the user models built by the different systems. In this section, we discuss the problem of user model semantic interoperability and exemplify some solutions for this problem provided by ontological technologies.

The task of interfacing two user models involves the resolution of modelling differences or discrepancies. There are four main sources of potential discrepancies between user models collected by different systems:

- different user modelling approaches
- mismatches in domain models
- different representations of user profiles
- scale discrepancies.

The following subsections briefly analyse the nature of these discrepancies and potential ways to solve them.

4.3.1 Integration of different user modelling approaches

As already mentioned, adaptive systems can represent information about their users in many different ways, including concept-based, keyword-based, stereotype, constraint-based, Bayesian networks, collaborative filtering, item-based, case-based, etc. (see Section 2.2). Several projects address the task of employing several user-modelling approaches in a single hybrid system to serve better adaptation (see Burke (2007) for a review of hybrid adaptive recommenders). However, the problem of translation between two fundamentally different representations of a user created by systems that understand only one of them is a great challenge. Ontologies can provide a solution in a limited number of cases. For example, ontology-based knowledge acquisition methods can be employed for transforming keyword-based models into concept-based; lexical ontologies, such as WordNet, can also facilitate the extraction of semantic information from keyword vectors. Suraweera et al. (2004) introduce the project ASPIRE that utilises ontologies in constraint-based tutors.

However, not all user-modelling approaches require representation of domain semantics, or semantics of user profiles. For example, the main component of user models for adaptive systems exploiting social filtering or item-based filtering is the matrix of interest ratings for items provided by the users. The translation of such models require completely different solutions, such as the ones discussed in Berkovsky et al. (2006), where the authors describe mediation between a case-based and a keyword-based models.

Here we do not analyse the problem of interoperability of non-semantic user models and focus on projects exploring the opportunities ontologies provide for interfacing information about the same user collected by different adaptive systems.

4.3.2 Resolution of domain discrepancies

Overlay user modelling uses a conceptual domain representation as a basis for structuring user information. However, the same domain (or related parts of intersecting domains) can be modelled differently in different adaptive systems. For example, Amazon.com and Blockbuster can model customer interests as overlays over ontologies of book and movie genres, correspondingly. Figure 15 demonstrate how similar parts of these ontologies can differ from one another.

- The same concepts can be named differently, for example, the concept label ‘Fairy Tale’ in the book genre ontology and the concept label ‘Fantasy’ in the movie genre ontology.
- The same concept labels can model different concepts. For example, the concept label ‘Fiction’ models different genres in both ontologies.
- The structure of inter-concept relations can be different. For example, the concept ‘Kids’ in the book genre ontology is a top-level category, while in the movie ontology is a sub-concept of the concept ‘Family’.
- The granularity of modelling can be different, e.g., several concepts in the movie genre correspond to the single concept ‘Fiction’ in the book genre ontology.
- The focus of modelling can be different, which may result in any of the discrepancies mentioned above. For example, the concept ‘Fairy Tale’ in the book ontology is a genre for Kids. At the same time, the movie genre ‘Fantasy’, the most relevant to ‘Fairy Tale’, is a sub-concept of ‘Fiction’ in the movie genre ontology. This discrepancy reflects a different view on the modelling of similar semantics. The book ontology stresses the strength of the fact that the majority of fairy tale readers are kids, while the movie ontology models the fact that Fantasy ‘is-a’ Fiction.

Finally, the representation formalisms used by the systems to express the domain model can be different. One system can store it in a relational database, yet another can externalise it as a set of XML files.
The domain discrepancies lead to different interpretation of identical user actions. For example, according to Figure 15, the interest model of a buyer of books ‘The Firm’ and “Harry Potter and...” will differentiate from the model of a user who rented movies adapted from the same books, although these actions indicate interests in similar content. While a user is modelled within a single system, her or his model is consistent with the underlying domain representation. However, when we try to merge two models to benefit of relevant user activities across the systems, domain discrepancies prevent from coherent interpretation of identical user actions performed with books and movies.

Ontologies provide a principle solution to the problem of domain model discrepancies. In our example, the presence of a publicly available ontology of genres represented in a sharable format and adopted by the developers of both systems would bring the user models of both systems to a common ground. Probably, the most successful initiative to develop a single domain ontology shared by a large community of experts is the Gene Ontology project (Ashburner et al., 2000). This ontology is a ready-to-use instrument for modelling truly consensual knowledge in the area of gene research. Although we are not aware of existing adaptive systems using this ontology as a domain model, the developers of future systems in that field should consider its employment.

In the area of user-adaptive systems, the UbisWorld project mentioned in Section 4.1.3 is an example of centralised ontology development. The set of UbisWorld ontologies models various aspects of ubiquitous computing for the development of adaptive ubiquitous applications understanding each other and providing consistent adaptation for the same user based on the shared knowledge about her or him. Another global ontology that can be useful as a mechanism of domain discrepancy resolution is WordNet. As mentioned before, several adaptive systems use WordNet to remedy such problems of keyword-based models as word polysemy and synonymy. If a single user has worked with several such systems, it should be possible to merge the WordNet-based user models of these systems to obtain a more complete profile of the user.

OntoAIMS is an example of integration of two separate adaptive systems, based on a shared domain ontology (Denaux et al., 2005b). Figure 16 demonstrates the OntoAIMS architecture. The OWL-OLM system supports interactive elicitation of learner knowledge (see Sections 4.1.2 and 4.2.5). Both the domain ontology and the elicited learner model are presented in OWL and available to the AIMS system, which provides the learner with adaptive access to available educational content.

**Figure 15** Example of domain discrepancies (see online version for colours)

![Genre](Image 85x598 to 238x751)

**Figure 16** OntoAIMS integrated architecture

*Source: Denaux et al. (2005b)*

AIMS structures the material as a hierarchy of tasks, where every task has a set of associated domain concepts. The model supplied by OWL-OLM is utilised by AIMS for adaptive recommendation of a next task to the learner (once, the learner masters all prerequisite concepts) and for support of adaptive browsing through the material related to the current task. While learners browse task-related material and definitions of relevant concepts in AIMS, both systems update their models as well. Hence, OntoAIMS provides full integration of two AES based on shared user model. A common domain ontology is used as a basis for user model mediation.

Most often however systems operating in the same domain rely on different ontologies. In such cases, ontology-mapping techniques can help in setting common understanding of the domain semantics. Once the mapping is found, the systems should be able to align their domain models and mediate users’ information. We are not aware of existing projects utilising general automatic ontology mapping techniques for translation of user models. A few papers considering the problem of cross-system personalisation based on domain ontology mapping either rely on manual mapping or assume the existence of a reference ontology that can help to map the top concepts of systems’ domain models.

An example of the first approach is the Medea learning portal (Trella et al., 2005). Medea provides students with access to multiple educational systems from a single point. To maintain consistent adaptivity across several applications Medea needs a mechanism for mediating user models between the applications. This task is achieved by manual mapping of the domain models of the adaptive systems to the central domain model of Medea. Hence, a student’s knowledge assessed by any adaptive educational system accessed through Medea is translated into Medea’s domain representation and stored in the central student model of Medea. If a system needs to obtain the current state of the student’s model to appropriately tailor its behaviour to her or his knowledge, it asks Medea. Medea retrieves the modelling information, translates it into the system’s representation and reports it to the system.
The same approach has been also implemented in the M-OBLIGE model (Mitrovic and Devedzic, 2004). The authors consider a problem of mediating knowledge models of the same learner between several intelligent tutors teaching different aspects of database management (entity-relationship diagrams, SQL, database normalisation). Although the domains of these tutors are different, they share several concepts. For reasoning across tutors’ local ontologies and mediating users’ knowledge models M-OBLIGE introduces an ontology processor. The ontology processor uses the global ontology (e.g., the general ontology of data models) to link tutors’ local ontologies and support cross-tutor personalisation.

Although these approaches potentially ensure high quality of user model mediation, they have serious shortcomings. Medea’s approach requires manual mapping (by a domain expert) of the central domain model to the model of a new adaptive system. Since the size of domain models can exceed several hundred concepts, the task can be very time- and expertise consuming and thus a non-realistic scenario for multiple systems for multiple domains. The M-OBLIGE framework assumes the existence of a global ontology linked to the local ontologies of the tutors. Such commitment is only possible in the settings of a close team of experts (instructors) working together. Ontologies and tutors developed by other teams will not comply with the M-OBLIGE requirements and will not be available for cross-personalisation. Since the automatic ontology mapping techniques have the potential to deal with the general task of solving domain model discrepancies, we expect new projects investigating this research direction to appear.

4.3.3 Resolution of user profile discrepancies

Adaptive systems can collect diverse information about users and maintain complex user profiles representing multiple aspects. Even though most of the users’ characteristics modelled by the existing adaptive systems are typical, the actual profile representation varies from one system to another. As an example, we consider the structure of the top categories from two standard learner profiles: IEEE PAPI (Figure 17) and IMS LIP (Figure 18).

PAPI’s learner profile has six top-level categories (IEEE-LTSC, 2001):
- **Personal info**: demographic information
- **Relations**: learners’ relation with other subjects of the education process (peers, teachers, etc.)
- **Security**: learners’ credentials and access rights
- **Preferences**: what objects the learners can and like to work with
- **Performance**: the results of learners’ assessments
- **Portfolio**: information about learners’ previous experiences.

The core categories of IMS LIP are (IMS, 2001):
- **Identification**: individual data such as names, addresses, contact information
- **Goal**: personal objectives
- **QCL**: qualifications, certifications and licenses
- **Activity**: records about education, work and service (military, community, etc.)
- **Interest**: descriptions of the hobbies and other recreational activities
- **Competency**: descriptions of the skills the learner has acquired
- **Accessibility**: cognitive, technical and physical preferences of the learner
- **Transcript**: academic performance of the learner with respect to a particular institution
- **Affiliation**: descriptions of the affiliations associated with the learner
- **Securitykey**: descriptions of the passwords, certificates, PINs etc.
• **Relationship**: definition of the relations between the other core data structures, e.g., QCL and the awarding organisation.

Even for these standardised profiles developed for similar purposes by large committees of established professionals from the same research field, the top-level categories greatly deviate one from another. Identical categories have been named differently (e.g., PAPI’s *personal* and IMS LIP’s *identification*). Some PAPI categories correspond to several IMS LIP categories (e.g., PAPI’s *relation* and IMS LIP’s *affiliation* and *relationship*) and some IMS LIP categories cover data that belongs to several PAPI categories (e.g., IMS LIP’s *activity* and PAPI’s *performance* and *portfolio*). Figure 19 represents the complete mapping of IEEE PAPI and IMS LIP as given in Klamma et al. (2005).

The discrepancies in user profiles need to be resolved for meaningful integration of user modelling information. Even though the list of potential discrepancies in user profiles will not differ much from the list of domain discrepancies, in practice, this problem appears to be easier than the general problem of matching domain ontologies, because of several reasons. From one side, the development of a user profile ontology is a more formal task, than the development of an arbitrary domain ontology; from another, the set of potential elements is fixed and well-structured, and the developers of user profile ontologies approach the design in a more accurate way, than an average domain ontology author. Besides, the number of existing user profile ontologies is smaller than the number of domain ontology author. Therefore, existing projects exploit one of two main approaches to the problem of interoperability of user profiles:

- development and maintenance of a central user profile ontology that acts as a common ground for multiple applications
- manual mapping of existing user profile ontologies by setting up conversion rules or build an integrated model.

The GUMO ontology is an example of the first approach (see Section 4.1.3). It has been built from scratch for the purposes of modelling a user in ubiquitous settings. Several adaptive applications model their users based on GUMO (Kruppa et al., 2005; Stahl et al., 2007).

As to the second approach, we have previously discussed several initiatives to merge existing learner profiles (Dolog and Nejdl, 2003; Jovanovic et al., 2006; Ounnas et al., 2006) (see Section 4.1.3 for details). The most cited work (Dolog and Nejdl, 2003) suggests a representation of learner profiles that inherits categories from both IEEE PAPI and IMS LIP, and extends them with new concepts. In Dolog and Nejdl (2007) the authors extend the previous work with several additional RDF models (document model, user activity model, etc.) and propose a unifying approach to the development of adaptive educational applications on the SW. All declarative and descriptive knowledge is represented in RDF. The adaptation strategies are formalised as sets of rules. Among the benefits of this approach are the high level of modularity, interoperability, and reusability of existing models and components. The development of a new system could be reduced to the development of a new domain ontology or formulation of a new set of adaptation rules.

### 4.3.4 Resolution of scale discrepancies

The final source of divergence between user models is the different scales used for assessment of users’ attributes. For example, two educational adaptive systems can use the overlay user modelling approach, operate in the same domain, and rely on the same domain ontology, yet they can model a user’s knowledge using different scales for knowledge assessment. Scales can be binary (knows/does not know), categorical (high/medium/low), numeric (from 1 to 5), and probabilistic (the certainty that the user knows the concept). Several problems can arise during the mapping of two scales:

- How to map two different categorical scales (when the category labels are different or the numbers of categories are different)?
- How to map a probabilistic scale into a categorical one (categories thresholds)?
- How to align the internal inference mechanisms of two different systems (one system can use an asymptotic formula for attribute value calculation, when another can use a Bayesian approach)?
Scale ontologies could be developed to help solving the first two problems. If numeric values are expressed using XML standard data types, they become available for standard RDF reasoning engines, which can be used to provide automatic mapping to ontology-based scales. If systems’ internal inference mechanisms are explicated and serialised in RDF, a set of rules could be developed to map the values inferred using different algorithms. We are not aware of existing projects resolving scale discrepancies; further research has to be done in this direction, since no practical solution for the problem of user model interoperability is possible without dealing with scale discrepancies.

Although the resolution of any type of discrepancies will lead, in practice, to some loss of modelling data, mediated user models stay a valuable source of information about users. In most cases, the alternative to a non-precise user model translated from another system is an empty user model, which deviates from the actual user’s state much more drastically.

5 Conclusion

This paper has attempted to bridge two research areas important for the future generation of web applications: user modelling and ontologies. Sections 2 and 3 have summarised the technologies developed in both fields and Section 4 has detailed the promising trends of how these technologies have been used together to build ontology-based adaptive systems. In the first sections, we have not tried to present the complete lists of existing approaches. Several cutting-edge user-modelling problems have not been mentioned, such as user modelling in virtual reality, privacy-enhanced user modelling, etc. Some topics of ontological research, while being actively discussed by the SW community, have been left out of the scope of this review (such as ontology reasoning and querying, ontology versioning, ontology validation and evaluation, ontology robustness and scalability, etc.). Instead of drawing the complete pictures of these two fields, we have chosen to describe those technologies that have the potential to demonstrate the applicability of ontologies for user modelling in modern AW applications. The choice of topics covered in the first sections was mainly driven by the existing projects exploring the “ontology-user modelling” synergy described in Chapter 4.

The discussed here direction of research is very dynamic. Most of the works referred in Section 4 have been implemented in the last five years. Many of the references report work in progress, which is still under development. New ideas have been generated and new papers have been published as we have been writing this overview. While we could hardly cover every single project applying ontologies for user modelling and adaptation, we have tried to mention the most interesting studies from the research teams that are likely to continue working in this direction.

As the field develops, we expect to see not only new systems implementing more effective and complete solutions for the problems described in Section 4, but also new trends addressing novel challenges and proposing principally different approaches. Some of these trends, which we believe will attract the attention of the research community in the nearest future, are given below.

5.1 Ontology-based adaptive architectures

The role played by ontologies in the adaptive system design, which is often reduced to the representation of a particular knowledge component, can be broadened at both the design-time and run-time. Guarino (1998) proposes several ways of wider employment of ontologies by information systems. Architecturally, ontologies can be used to represent any knowledge component in the system (user model, domain model, adaptation model, communication model). This can lead to better reusability and interoperability of systems’ components, unified design patterns, and standardisation of communication protocols. The cross-model knowledge processing in the system can be entirely based on standard inference engines, which will make it consistent, effective and transparent for the third parties willing to reuse the system’s components.

Several research teams are working on the development of ontology-based adaptive architectures. Personal Reader is an architecture for adaptive support of reading web content, based on ontologies and SW services (Henze and Herrlich, 2004). The Omnibus project is aimed at developing a unified ontology of learning, instruction, and instructional design, which provides a framework for full-scale ontology-based authoring tools for ITSs (Mizoguchi et al., 2007). The mentioned UbisWorld project utilises a comprehensive set of ontologies to implement a framework for interoperable ubiquitous personalisation (Heckmann, 2006).

5.2 Social semantic adaptive web

A recent trend in the web research focuses on connecting the SW technologies for representing, retrieving and processing knowledge with the interactive social interfaces of Web 2.0. The lack of content in the SW applications and the lack of structure in the community-authored content of Web 2.0 call for a cooperative solution. The common approach of multiple projects originated in this area is to exploit the large body of raw content and metadata produced by the users of social web applications, while employing the structural knowledge and standard representation technologies offered by the SW. Although it often comes at the price of formalisation and richness of the resulting knowledge models, the practicality of this approach is promising. Many Web 2.0 applications
merged with SW technologies obtain new functionalities and/or new ways of use, for example, semantic wikis (Auer and Lehmann, 2007; Buffa et al., 2008), semantic community web portals (Gruber, 2008; Staab et al., 2000), semantically-enhanced tagging facilities (Jaschke et al., 2008; Newman et al., 2004), etc.

Adaptive technologies have successfully made their way to the Web 2.0 platform (e.g., collaborative recommendation (Konstan et al., 1997), social navigation (Farzan and Brusilovsky, 2006), community-based search (Smyth et al., 2005)). This review gave multiple examples of adaptive applications implemented by using SW approaches. Merging all three paradigms together should result in an improvement of existing approaches and, maybe the emergence of new ones. For example, hybrid recommender systems, implementing both collaborative and content-based adaptation are known to outperform the pure collaborative recommendation for a number of tasks. However, the content-based component is very much dependant on the quality and richness of content models. Implementation of such models with the help of ontologies could improve the content-based adaptive inference, as well as enhance the population of these models through ontology mapping, ontology-based annotation and ontology learning techniques.

Social navigation, while being an efficient technology for automatic open-corpus content quality control, does not implement any mechanisms for navigating the individual user to the best document at the best moment of time. Semantic technologies can help to ‘personalise’ socially produced navigational cues through maintaining richer models of users and content.

One of the biggest challenges for community-based adaptive search is to identify the right community, which the user is currently in, in order to tailor the search results appropriately. Ontologies could help to draw stricter borders between the communities by enriching their profiles with semantic information. Currently these profiles are populated mainly based on the tags and queries used by their members. Such projects as WordNet and Tag Ontology (Newman et al., 2004) can help community-based systems to migrate from lexical to semantic models.

A recently launched project called Twine is a working example of a joint application of semantic, social and adaptive technologies (Radar_Networks, 2008). Twine is a semantic bookmarking service, which allows users to share information they have discovered on the web, provides automatic semantic analysis and markup of the shared web-pages and even generates adaptive recommendations based on the individual user’s history and resources submitted by others.

There are more examples of potential synergy between these areas of web development. We expect more research and commercial projects to emerge in the nearest future.

Acknowledgement

We would like to express our gratitude to Peter Brusilovsky and Michael Spring for their help in preparing this review.

References


Dicheva, D. and Dichev, C. (2007) ‘Helping courseware authors to
Dewey, M. (1876)
Denaux, R., Aroyo, L. and Dimitrova, V. (2005a) ‘An approach for
Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L.,
De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B.,
Denaux, R., Aroyo, L. and Dimitrova, V. (2005a) ‘An approach for
Ontological technologies for user modelling


Notes

1 The query has been tossed on 20 August 2009.
2 Some researchers also use the terms ‘ontology alignment’ and ‘ontology merging’ as synonyms of ‘ontology mapping’.
3 These are not real ontologies used by Amazon.com and Blockbuster. We have designed them to demonstrate typical discrepancies that can occur between two models of the same domain.