A Hybrid Approach to Identifying User Interests in Web Portals

Fedor Bakalov[1], Birgitta König-Ries[1], Andreas Nauerz[2], and Martin Welsch[2]

[1]Friedrich Schiller University of Jena, Institute of Computer Science
Ernst-Abbe-Platz 1-4, 07743 Jena, Germany
{fedor.bakalov|birgitta.koenig-ries}@uni-jena.de

[2]IBM Research and Development
Schönaicher Str. 220, 71032 Böblingen, Germany
{andreas.nauerz|martin.welsch}@de.ibm.com

Abstract: Web portals pioneered as one of the earliest adopters of personalization techniques to help users dealing with the problem of information overload. Nowadays they are extensively used as a single-point of access to the vast amount of resources available on the Web and in enterprise intranets. A number of researchers have been investigating the possibilities to enable portals to deliver the content in a highly-personalized manner in order to provide users with a quick and efficient access to the subset of resources relevant to their information needs. However, in order to achieve such a personalization effect, the portal needs accurate and up-to-date information about users, especially the information about their interests. In this paper, we describe a hybrid approach to identifying user interests in Web portals. In our approach, the portal is enabled to "learn" the user interests from the content of visited pages. In addition, it is empowered to provide users with an open access interface to their user models to let them explicitly specify their interests and, in case of incorrectly identified interests, outvote the portal.

1 Introduction

Web portals emerged in the late 1990s primarily as the gateways to different information resources available on the Internet. Companies like AOL, Altavista, and Yahoo used them to guide their user communities through the network. The first portals provided users with Web directories, fulltext search capabilities, and certain communication services like email and chat. Later, portals gained special attention among enterprises as a platform to integrate not only the corporate information resources, but also the company's legacy systems. Nowadays, a large number of organizations use portals extensively as a complete e-business solution providing users a single-point of access to the vast amount of company resources and applications. Furthermore, with the advent of Web 2.0, portals have gained popularity as the gateways to community-driven resources like wikies, blogs, mashups, and many others.

However, with the constantly expanding growth of information available through Web portals, it has become more difficult and time consuming to find relevant resources. In

today's Web 2.0 this problem has become even more prominent due to the large number of different users contributing. In case of a portal consisting of several hundred pages with contributions from different members of a community, traditional hierarchical navigation is no longer efficient as it is not feasible for the administrator to come up with one optimal topology that would fit to the navigation patterns of every individual user. Furthermore, it is unlikely that users will place the content that they contribute according to some plan an administrator drew up. Thus, the portal industry is facing a challenging research question - how to make portals adaptive to the information needs of individual users.

A number of researchers have been investigating the possibilities to create individual personalized information spaces, where the user could easily access the subset of resources that are relevant to his/her information needs. However, in order to create such a personalized information space, the portal needs to "know" certain information about the user. Such information is usually stored in a so called *user model*. Brusilovsky and Millán defined user model as *"a representation of information about an individual user that is essential for an adaptive system to provide the adaptation effect, i.e., to behave differently for different users"* [BM07]. Depending on the desired adaptation effect, the user model may contain such information as user interests, background, goals, traits, etc.

In this paper, we describe a hybrid approach to identifying user interests in Web portals. The goal of our research is to enable the portal to "learn" the user interests from the content of visited pages. In addition, we aimed to provide users with an open access interface to their user models to let them explicitly specify their interests and, in case of incorrectly identified interests, outvote the portal. The remainder of this paper is organized as follows. Section 2 and Section 3 provide details on the user model and domain model respectively. In Section 4, we elaborate on the approach itself and describe the steps involved in the user modeling process. In Section 5, we describe the prototypical implementation of our approach and provide results of a preliminary evaluation. Section 6 provides a short overview of the previous research that has been done in the area of user modeling. Finally, Section 7 concludes the paper and outlines the directions for our future work.

2 User Model

In our approach, the user model logically consists of two parts: a static part and a dynamic one. The static part contains the time invariant information about users: date of birth, gender, first language, etc. This is the information that users provide explicitly when they register in the portal. In the dynamic part, we represent the constantly changing user features, namely user interests.

Basically, we define user interest as a fact indicating that a given user is interested in a certain term with a certain degree of interest. Here, the term is a reference to a concept denoting either a real world object, like company, geographic location, person, etc., or an abstract notion, like an area of science, discipline, technology, etc. The concepts themselves are stored in the underlying *domain model*, which is represented as an ontology providing machine-processable semantics of the contained entities (see Section 3).

The degree of interest denotes the extent to which the user is interested in a given term. We distinguish three levels of interest and identify them by the following linguistic variables: *interested*, *partially interested*, and *not interested*. Also we introduce an auxiliary linguistic variable *blocked* to mark the terms that were explicitly blocked by the user. This information could be used by the portal to stop tracking user interest in the given term as well as by the personalization engine to stop recommending resources about that term. Section 4.3 explains in detail the methods for determining the degree of interest.

Following the research described in [Sch06], we model the user interests as time dependent features. We assume that a user might be interested in a certain term only for a certain period of time. For instance, a football fan is probably interested in the World Cup mostly during the time period when the event takes place.

Thus, the interest user model is represented as a collection of tuples *(U, T, I, V)*, where:

- *U* is the user portal ID
- *T* is the URI of an instance from the domain model
- *I* is the linguistic variable indicating the degree of interest
- *V* is the time period of the interest validity

For instance, the fact denoting an interest of a fictional user Klaus in the World Cup 2010 can be represented as: *(klaus, http://www.minerva-portals.de/domain-model.owl#WorldCup2010, interested, (2010-06-01, 2010-07-31))*.

3 Domain Model

A domain model is a data model that defines concepts in a given domain, e.g., chemistry, medicine, biology, etc. [GPFLC03]. We have chosen the finance domain for our proof-of-concept implementation. Therefore, in our domain model we define the concepts that users from the financial realm may work with, such concepts as stock, bank, account, etc.

The domain model is represented as an OWL ontology[1], which defines the domain concepts as ontological classes by specifying their properties and relations to other classes. E.g., an event *Acquisition*, denoting the fact of acquiring one company by another one, is defined as a subconcept of *FinancialTransaction* and is described by such attributes as *date*, *acquiree*, *acquirer*, and *transationAmount*. The ontology is grounded on the Proton Upper Module[2], which defines upper-level concepts, such as organization, location, person, etc. Under these concepts, we define finance-specific terms partially reused from two existing finance ontologies, namely LSDIS Finance Ontology[3] and XBRL Ontology[4].

[1] http://www.minerva-portals.de/finance-domain.owl
[2] http://proton.semanticweb.org/
[3] http://lsdis.cs.uga.edu/library/resources/
[4] http://xbrlontology.com/

The domain model also contains instances of concepts. E.g., for the concept *Company*, we specify such instances as "Microsoft", "IBM", "Google", etc. Class instances are required to represent specific user interests in the user model. Inclusion of new instances into the ontology is performed automatically using the Calais service (see Section 4.1). We harness the service in order to extract named entities (such as company, industry term, technology, etc.) from the text of documents accessed by users. The extracted entities that are not yet present in the domain model are then inserted into the ontology as instances of the corresponding concepts.

4 Approach

Our approach to identifying user interests involves the following activities. First, the portal collects the terms indicating user interests into the user model by analyzing the content of visited pages as well as by allowing users to explicitly enter them through a special interface. Second, the collected terms are semantically enriched by refering to the corresponding instances in the underlying domain model. Finally, for every collected term, the portal determines the degree of interest either by leveraging the term frequency, or semantic relation among the terms, or by letting the user specify it explicitly.

4.1 Collecting Terms

In our approach, we distinguish two sources for collecting new terms into the user model. First, we allow for automatic extraction of terms from the pages visited by the user. Second, we provide the user a possibility to manually enter new terms into his/her user model. For the *automatic* extraction of terms, we leverage the Calais Web service[5]. Calais is an unstructured text analysis service, which can receive an HTML or plain text document as an input and return an annotated document in RDF[6] format. More specifically, the service performs named entity recognition: it extracts certain general and business-related entities such as company, location, person, etc. It also supports extraction of certain events and facts, such as acquisition, bankruptcy, family relation, etc. We developed a special portlet that analyzes the content of pages visited by the user with the Calais service and uses the extracted entities to update the domain and user models. For every extracted entity, the portlet checks if the domain model contains a matching instance. If it does not find one, it inserts a new domain instance using the information about the extracted entity provided by the service, which includes the entity label, full name, and type. Afterwards, the portlet makes an entry in the user log where it specifies the URI of the domain instance and the number of occurrences in the document. Finally, the portlet determines the degree of interest in the term (see Section 4.3) and inserts a new user interest into the user model. In addition to the automatic extraction of terms, we empower users to *manually* enter new

[5] http://www.opencalais.com/
[6] http://www.w3.org/RDF/

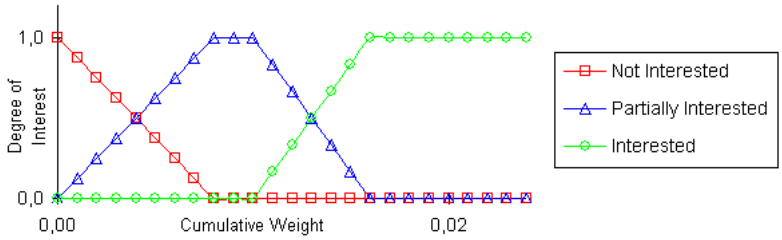

Figure 1: Fuzzy sets representing degree of interest

terms. We developed a user modeling portlet (see Section 5) where the user can access the terms stored in the underlying domain model and insert the terms of interest into his/her user model.

4.2 Determining Semantics of Terms

As mentioned above, the semantics of the existing user interests is represented in the underlying domain model. In case when the user manually inserts a new interest, the portal already "knows" its semantics because it is an existing domain instance. In the current implementation of the approach, we allow users to select new interests only out of the existing domain instances. However, we are currently investigating possibilities to enable user communities to collaboratively edit the portal domain model by adding new concepts and instances as well as connecting them through user-defined relations. Thus in the future, we plan to enable users to specify new interests by adding a new ontological instances directly into the domain model.

In case of new terms extracted by the Calais service, the portal needs additional information to determine the semantics. For this purpose, we developed a mapping between the Calais types and the concepts stored in the domain model. For every Calais type, we identified a corresponding domain concept and mapped it to the Calais' one through the OWL *sameAs* property. For instance, domain concept *StockIndex* is mapped to the Calais type *MarketIndex* through the following assertion: *http://www.minerva-portals.de/finance-domain.owl#StockIndex owl:sameAs http://s.opencalais.com/1/type/em/e/MarketIndex*. All new terms coming from the Calais service are inserted into the domain model as instances, which classes are determined using the mapping. E.g., using the above mentioned mapping, term FTSE 100[7] will be inserted as an instance for the domain concept StockIndex.

4.3 Determining Degree of Interest

As described in Section 2, we distinguish three levels of user interest and identify them by the following linguistic variables: *interested*, *partially interested*, and *not interested*. Every variable is associated with a fuzzy set, which is defined by the corresponding membership function [Zad65, Kav04]. The membership functions are based on *cumulative weight*, a real number that can take values from 0 to 1, that denotes importance of a certain term for the user with respect to the other terms. We use this value to define the membership functions that represent the degree of user interest; μ_{ni}, μ_{pi}, μ_i show the degree to which the user is *not interested*, *partially interested*, and *interested* in the given term (Figure 1). The cumulative weight is a constantly changing value and can be determined with one of the following three methods: log-based updates, inference-based updates, and user manual updates.

Log-based updates are performed automatically by the portal using the user log that stores occurrences of the terms extracted from visited pages. For every term in the user model, the portal calculates the term frequency value:

$$TF_{i,j} = \frac{t_{i,j}}{\sum_k t_{k,j}}$$

where t is the number of occurrences of $term_i$ for $user_j$, and the denominator is the total number of occurrences of all terms registered for $user_j$. The computed term frequency value of $term_i$ is specified as the term's cumulative weight, unless the user has already manually specified his/her interest in the term.

Inference-based updates leverage the semantic relations among the instances in the domain model. This is when the portal identifies new interests by propagating interest from the terms for which the user model already contains information about (e.g. the user has explicitly specified interest degree or it has been determined by the portal based on the term frequency). For instance, if the user model contains a user interest in Berlin and in the domain model Germany is connected to Berlin through the property *hasCapital*, then the inference engine can propagate user interest from the former to the latter. The taxonomical relations can be leveraged for interest propagation as well: interest can be propagated from a child concept to its parent and vice versa.

User manual updates are performed by the user explicitly. We developed a user modeling portlet (displayed in Figure 2) through which the user can access and edit his/her user model. The user can check the interest status generated by the portal and in case of incorrectly identified interest, the user can outvote the portal by manually changing the status, which will also affect the cumulative weight value. If the user *promotes* a term (e.g. from not interested to interested), the cumulative weight of that term will be increased up to the lowest cumulative weight value in the fuzzy set of "interesting" terms. Whereas, if the user *demotes* a terms (e.g. from interested to not interested), the cumulative weight will be set as the highest cumulative weight in the fuzzy set of "not interesting" terms. In case the user *blocks* a term, the cumulative weight of that term will be set to 0.

[7] Financial Times Stock Exchange Index

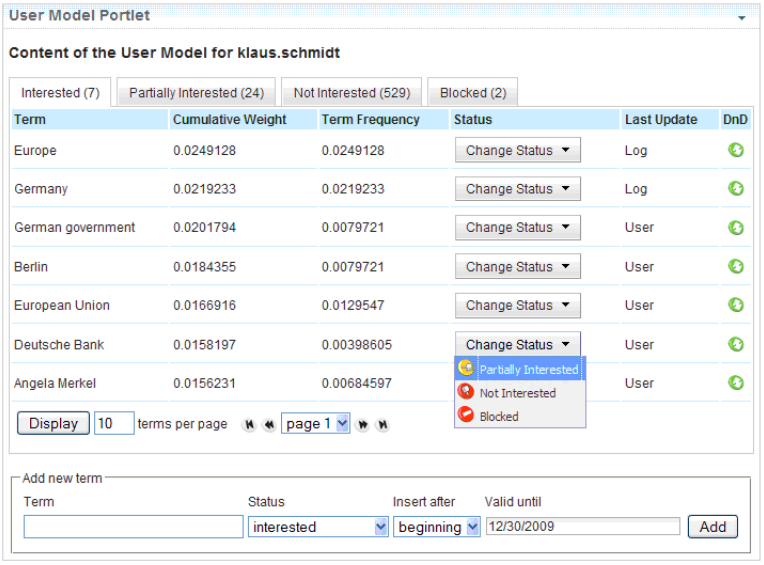

Figure 2: User modeling portlet

5 Implementation and Preliminary Evaluation

The approach described above has been prototypically implemented in IBM's WebSphere Portal. Figure 3 illustrates the system architecture of the prototype. The *portlet application* consists of three portlets. *Content portlet* provides users with news harvested via RSS feeds from news websites like BBC and CNN. The pages displayed in the content portlet are processed with the *models update portlet*, which sends the content to the *Calais service* and based on the extracted entities updates the user and domain models. Finally, the *user model portlet*, displayed in Figure 2, allows users to view content of their user models, add new interests, and change interest status of existing terms. Additionally, the portlet allows users to reorder the terms within the same interest group using the drag-and-drop technique.

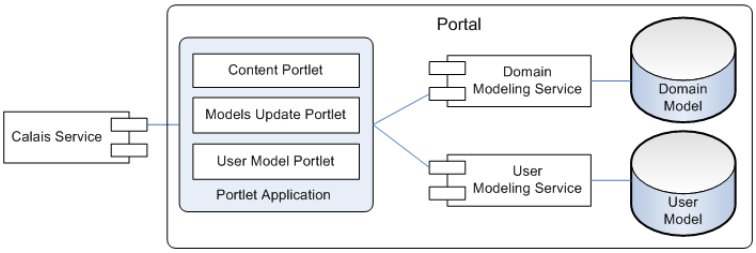

Figure 3: System architecture

The *user model* is implemented as a relational database. It stores information about user interests and logs containing the user browsing history and user model updates. The *domain model* is implemented as an RDF triple store deployed in the Sesame Framework[8]. Access to the user model and domain model is provided through user modeling service and domain modeling service respectively. The *user modeling service* provides such operations as, adding new log entry, performing log-based updates and user manual updates, and providing user interest list. Whereas, the *domain modeling service* is used for adding new domain instances and getting information about the existing instances, such as main label, aliases, and class.

The prototype has been evaluated with three fictional users representing three countries, namely, Germany, Russia, and France. Every user used the *content portlet* to read news related to the country he or she is representing. The content of visited pages was processed with the *models update portlet*, which created three corresponding user models. Table 1 shows the results of experiment.

User	News Topic	Visited Pages	Total Terms	Interests		
				Interested	Part. Interested	Not Interested
Klaus	Germany	381	638	5	30	603
Dmitry	Russia	72	226	8	59	159
Isabelle	France	14	45	28	17	0

Table 1: Results of experiment

We compared the three generated user models and identified the following issues. As the reader can see from Figure 4, the proportion of interests depends on the size of the model. Here the user model of *Klaus*, the largest model, contains less than 1% terms that the user is *interested* in, whereas in the user model of *Isabelle*, the smallest model, the number of terms that the user is interested in exceeds 62%. This happened because the functions determining the membership of a term in the interest fuzzy sets are based on the cumulative weight value, which is by default set to the term frequency, the value that depends on the size of the term set. Therefore, the membership functions must be defined for every user individually based on the size of the corresponding user model. We are currently investigating possibilities to make the fuzzy sets floating and adaptive to the size and content of the individual user models. We also aim to enable users to manually adjust the fuzzy sets, by which they can increase or decrease the number of terms in a certain interest group.

6 Related Work

The user model is an essential component for any system that aims at adapting content to the users' specific needs. A number of proposals have been made to identify and represent knowledge about users. Crabtree and Soltysiak [CS98] describe an approach to deriving user interests automatically by monitoring various user activities, such as reading documents, writing emails, and browsing websites. The resulting user model is represented as

[8]http://www.openrdf.org/

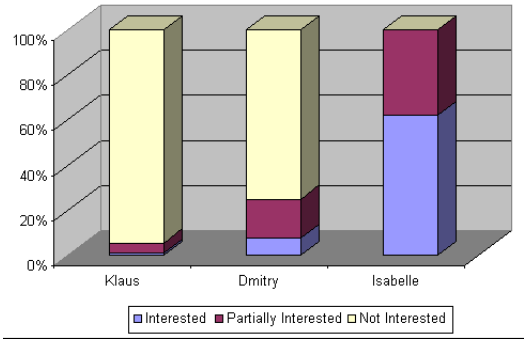

Figure 4: User interest statistics

a vector of weighted keywords denoting user interests. In [AHNJ07], Achananuparp et al. describe how the vector user models can be semantically enhanced. They propose using WordNet[9] lexical database to establish the semantic relations between the keywords in the user model.

Ontologies have gained a large interest as a means to semantically represent knowledge about users. Semantic relations among concepts in the user model allow deriving certain information about users that was not explicitly defined in the model. For instance, ontological represetnation of user interests enables propagation of the interest from child concepts to their parents as well as among similar concepts. In [Sch08], Schmidt describes ontology-based conceptual models that can be harnessed by learning management systems in order to provide users with the learning content tailored to the level of their knowledge as well as their situational needs. In [ZSS07], the authors elaborate an architectural solution that can automatically derive ontological user models based on Web content and user logs. A similar approach is described by Costa Pereira and Tettamanzi in [PT03].

An important aspect of user modeling is the ability to identify and represent the degree that the user is interested in a certain concept or possess knowledge on it, which in its turn can affect the quality of adaptation. In [Kav04], Kavcic describes a novel approach to representing the degree of user knowledge using fuzzy set theory. John and Mooney describe a similar approach to represent user interests in [JM01].

7 Conclusions and Future Work

In this paper we have presented a hybrid approach to identifying user interests in Web portals. We empowered the portal to harvest user interests in a non-intrusive manner by analyzing the content of visited pages and extracting semantic entities from them, which are then used to update the user interest models. Additionally, we have proposed an open interface to let the users explicitly specify their interests and, in case of incorrectly iden-

[9] http://wordnet.princeton.edu/

tified interests, outvote the portal. Our approach has been prototypically implemented in IBM's WebSphere Portal.

The information contained in the user model can be used to adapt the portal to user needs. We have showcased this with a less sophisticated user model in our previous work [NBKRW08]. In the future, we intend to build on that work to provide truly personalized portals. We will also extend the user model proposed here by leveraging community interests as well as individual user's interest. Beyond that, we will focus on evaluation of our ideas in a real-world setting.

Acknowledgements and Trademarks
This work has been done in the framework of the Minerva Portals Project funded by IBM Deutschland Research & Development GmbH.

References

[AHNJ07] Palakorn Achananuparp, Hyoil Han, Olfa Nasraoui, and Roberta Johnson. Semantically enhanced user modeling. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 1335–1339, New York, NY, USA, 2007. ACM.

[BM07] Peter Brusilovsky and Eva Millàn. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, chapter 1, pages 3–53. Springer, Berlin, Heidelberg, 2007.

[CS98] Barry Crabtree and Stuart J. Soltysiak. Identifying and tracking changing interests. *International Journal on Digital Libraries*, 2(1):38–53, October 1998.

[GPFLC03] Asunćion. Gómez-Pérez, Marianno Fernández-López, and Oscar Corcho. *Ontological Engineering*. Advanced Information and Knowlege Processing. Springer, 2003.

[JM01] R. I. John and G. J. Mooney. Fuzzy user modeling for information retrieval on the World Wide Web. *Knowl. Inf. Syst.*, 3(1):81–95, 2001.

[Kav04] Alenka Kavcic. Fuzzy user modeling for adaptation in educational hypermedia. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(4):439–449, Nov. 2004.

[NBKRW08] Andreas Nauerz, Fedor Bakalov, Birgitta König-Ries, and Martin Welsch. Personalized recommendation of related content based on automatic metadata extraction. In Marsha Chechik, Mark R. Vigder, and Darlene A. Stewart, editors, *CASCON*, page 5. IBM, 2008.

[PT03] Ce'lia Da Costa Pereira and Andrea Tettamanzi. An Evolutionary Approach to Ontology-Based User Model Acquisition. In Vito Di Gesu', Francesco Masulli, and Alfredo Petrosino, editors, *WILF*, volume 2955 of *Lecture Notes in Computer Science*, pages 25–32. Springer, 2003.

[Sch06] Andreas Schmidt. Ontology-based user context management: The challenges of dynamics and imperfection. In *in On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. Part I., ser. Lecture*, pages 995–1011. Springer, 2006.

[Sch08] Andreas Schmidt. Enabling Learning on Demand in Semantic Work Environments: The Learning in Process Approach. In Jo"rg Rech, Bjo"rn Decker, and Eric Ras, editors, *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*. IGI Publishing, 2008.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

[ZSS07] Hui Zhang, Yu Song, and Han-Tao Song. Construction of Ontology-Based User Model for Web Personalization. In *User Modeling 2007*, pages 67–76, 2007.