*Maciej Gawinecki*

# Modelling User on the Basis of Interactions with a WWW Based System

Summary of the Master Thesis

## Subject of the thesis

User modeling is a well-known technique for delivering personalized and relevant information to the user. In this work I discuss how user modeling can be utilized in the context of the Semantic Web. More precisely, I propose a complete process of construction and management of user models (profiles) for an agent-based travel support system. The problem has been inspired by the following scenario: *„Hungry foreign tourist arrives to an unknown city and seeks a nice restaurant serving cuisine that she likes. Internet, searched for advice about restaurants in the neighborhood, recommends" mainly establishments serving steaks, not „knowing" that the tourist is a fanatic vegetarian."*. Utilization of personalization of recommendation can solve this problem, by introducing user profiles (models) into the system. Overall, a user profile is a specific conceptualization of what user needs, is interested in, what are her skills, limitations etc. Following the given scenario, having a profile with negative opinions about eating meat the system would recommend only vegetarian restaurants. To provide the most relevant and actual recommendation user profiles should be updated with respect to changing needs/interests of the user. This process is called profile learning and is usually based on analysis of user reactions and choices during interaction with the recommender system. Building general recommender system is task consisting of: (a) defining how user profile is represented; (b) how it is initialized; (c) how and what part of user feedback should be tracked; (d) how this information can be used to update user profile; and (e) specifying algorithm of profile exploitation (the way the system filters recommendations on the base of the profile). The whole process is understood as user modelling (in the domain of recommender systems).

Since user modelling is not a new science and its utilization for personalization is not an innovation, the challenging issue was to adapt established methods of user modelling to functional and non-functional requirements of a specific travel support system. Particularly:

1. **Profile representation.** Since travel ontology is the centerpiece of the system it is naturally to represent user profiles as sets of statements about the "world of travel". The question is how to represent it in practice?
2. **Tracking user feedback.** The application has a classical, primitive WWW (or WAP) interface. The question is what kind of user reactions can be tracked and what information about user preferences do they carry?
3. **Profile learning method**. What profile learning method can be used in the case of such a feedback?
4. **Cold start problem.** Every new user does not have a profile in the system, so it cannot enjoy personalization facilities of the system. It requires a user to interact with the system for a long time to learn profile about her. The question is how such a profile can be initialized? Similar problem refers to adding a new object, that can be recommended, e.g. New Spaghetti Restaurant on Giovanni Street. Since there are no statements about it in any user profile, indicating that this object was never of user interest, it may never be recommended.
5. **Profile exploitation.** The problem is how a user profile should be exploited, so the system's response would contain only those travel objects, which both matches a user query and which features are of significant interest of a user (defined in her profile).
6. **Infrastructure.** Solution to all mentioned above aspect should fit into existing infrastructure of Travel Support System. Particularly, original system, constructed by Pawel Kaczmarek [14,15], posed as Content Management System, realizing idea of Controller-View-Model design pattern with utilization of agents. More precisely, it consisted of the following agents:

(1) *Proxy Agent* – transducer between browser-oriented HTTP interface and agent-based system, (2) *Session Handling Agent* playing the role of a Controller, (3) *View Transforming Agent* behaving as a View transformer by utilization of a Raccoon server, (4) *Personal* Agent, one for each logged user, playing the role of a Model generator, and (5) *Restaurant Service Agent* as a data source of ontology-demarcated restaurant data. To realize user modelling functionalities required extending existing infrastructure so it could: (a) identify each user in the system; (b) track user behaviour; (c) store user profiles; (d) change user profiles, i.e. (d1) initialize is on the base of data provided by a new user, and (d2) learn user profiles on the base of users' history; (e) personalize recommendations, provided by *Personal Agent*, i.e. exploit a user profile; (f) prompt a user with question necessary to gather explicit feedback about visited restaurants. Main questions here were: (I) how to distribute these responsibilities to separate agents to find compromise between legibility of abstraction and communication performance, and (II) how to bring together user-directed navigation and proactivity of *Personal Agent* (particularly functionality of asking questions), i.e. solve problem of mixed initiative in the existing system.

**Research methodology**

Initial reading of relevant literature has shown that user modelling is an extensive domain and there is a need for a clear taxonomy of recommender systems. The work of Montaner [10] responds to this problem very well as it contains classification of existing recommending systems on the base of methods each of them uses for each phase of user modelling. Lots of support I received also from members of user modelling mailing list[1]. Particularly, George D. Magoulas from University of London provided me with *the glossary of terms* to the book *„Adaptable and Adaptive Hypermedia Systems"* [7], which very clearly precises definitions of such terms as user modelling, personalization, recommender system etc. Knowing that, I could conduct and systematize my research according to the order of aforementioned problems.

1. **Profile representation.** It has been said, recommended objects (restaurants) are demarcated as instances of restaurant ontology (extracted from the Chefmoz project [3,4]). Therefore I was looking for such a representation of a user profile, which allows to define user opinions not only about particular restaurants, but also about their features (such a served cuisine, dress code allowed etc). Such a representation allows for utilization of feature-based filtering method, which returns not only those objects, about which similar user has asked, but also those with similar characteristics. The inspiration for a solution raised from profiles modelling knowledge of students (i.e. students models), which has been utilized in e-learning systems [9]. In comparison to TSS, the knowledge in such systems has been represented by structures simpler then ontologies, i.e. taxonomies. In my solution probability-like measure has been applied to each statement in domain ontology models and thus it strengths of user preferences. Since restaurant ontology has been demarcated in RDF and because RDF provides such tools as (a) referencing to any resource in the ontology and (b) reification (creating statements about statements) it became natural to represent user profiles also in RDF.
2. **Tracking user feedback**. Type of user interface (here WWW) has substantial impact on the type of feedback collected by the system, and so – indirectly – on the choice of relevant profile learning method. However, the WWW interface brings limitations to observing user behaviour, especially when talking about possibility of observing negative feedback. It is often not possible to tell it apart from the fact of lack possibility to read (e.g. because the information did not fit inside of a browser's frame). Therefore recommender system based on such an interface should collect only tokens of positive of interest [13]. It should also collect both implicit and explicit feedback, whereas the latter one should be considered as more important [12,13]. However, the user should not be pushed to provide explicit

1 <um@di.unito.it>. The aim of this list is sharing messages and news relevant to the research and development activities in the User Modeling and User-adaptive systems areas.

feedback, but she should be allowed to do that. There I decided to log information about keywords that appear in user queries, URIs of restaurants that user clicked and positive answers to the question: *„Have you enjoyed restaurant X ?"* if the user decided to response to it.

3. **Profile learning method**. The most difficult problem was to find such a profile learning method, which could utilize the limited amount of feedback. Here, I have been inspired by the work of Fink and Kobsa [16]. The authors utilizes techniques of univariance significance analysis based on frequency of occurrence of only positive user feedback. Moreover, tourist objects, which can be recommended by their system, are originally categorized into domain taxonomy (e.g. castles and churches belongs to the architecture category). I have adapted their approach by using RDF-demarcated ontology to represent domain instead of a taxonomy structure. This resulted in feature-based learning method, since my ontology not only classify objects possible to recommend, but also describes each each of them in a form of features' list. During this work I have been supported by Alftred Kobsa, one of the authors of this method and by dr Waldemar Wolynski from Department of Probability and Mathematical Statistics of my Faculty.
4. **Cold start problem.** Both introducing a new user into the system and a new object to be recommended is called cold-start problem. In case of a new user, her profile can be initialized by use of one of the following methods: manual edition of profile, training set and stereotyping [10]. The latter became my subjective choice. Stereotyping is a general method for classifying users into categories and then making predictions based on stereotypes associated with particular categories. Each stereotype contain typical assumptions that one makes about members of that category [6]. When defining such a method one need to answer to the following questions: (a) How many and what stereotypes should be defined ? (b) Which attributes of a new user should be used a base for prediction about the most matching stereotype(s) ? (c) How similarity between a new user and existing stereotypes should be defined ? (d) How opinions from the most similar stereotype(s) should be translated into a user profile ? (e) In what a way the user can be questioned about her attributes ? Of course, the first question is the key one. Originally I assumed that restaurants collects information about types of clients they serve. Such data would allow me to match attributes of these clients with attributes of their favourite restaurants. Unfortunately, all my consultants (Joanna Ochniak from the Academy of Hotel Management and Catering Industry in Poznan, dr hab. Andrzej Maslowski from the Institute of Home Market and Consumption and dr George Lekakos from Department of Management Science and Technology of Athens University of Economics and Business) confirmed, there such data are collected only by huge restaurant chains and only for internal use. Publicly accessible information allows only to predict general behaviour of consumers (e.g. pairs without children eat outside more often than families). An alternative approach would be constructing stereotypes on the base of results of questionaire conducted among staff of restaurants in my city. After consultation with dr hab. Michal Chmara from the Institute of Sociology of Adam Mickiewicz University in Poznan it appeared that conducting such a survey with respect to principles of the art is a problem for a separate master thesis. Therefore I constructed stereotypes manually, on the base of my prejudices. More precisely, since user profiles describes their owners by set of opinions and personal data (age, wearing style, richness and profession), therefore stereotypes are described by set of strong radical opinions and vectors of matching personal information. Similarity between personal information provided by user and those belonging to stereotypes is computed and the most matching stereotype is chosen as a resource for a new user's opinions. Since personal data are of different type (nominal, ordinal and interval types), I have adapted type classification approach from data clustering domain [11] to define similarity measure.
5. **Profile exploitation.** Since neither Kobsa nor Fink mention in their work, how a user profile could be utilized to rank recommended objects (i.e. no profile exploitation method is proposed), therefore I developed my own one. Rank of each travel object is called here

*temperature of interest* and represents the "probability" that a given object is a favourite of the user. To calculate the temperature of a travel object (called *active object*) two aspects of the situation have to be taken into account. First, features of the active object. Second, user interests in them, represented in the user-profile – if a given feature has no preference specified then it cannot be used. Objects, which temperatures are below certain threshold are filtered out and the remaining ones are sorted in respect to their tempeature in a descending order.

6. **Infrastructure**. In the previous section I have mentioned functional requirements for extending established agent-based infrastructure. It becomes naturally to wrap these functionalities with agents (see Figure 1-4). Particularly, I introduced *Profile Managing Agent*, which stores and provides access to user profiles, initializes and learn them. More precisely, open source Jena semantic framework (strongly supported by Hewlett Packard) has been used to process RDF data and store them in persistent database models. Data for learning process are provided by *Session Handling Agent*, which has been extended with ability of tracking user behaviour. This became a natural choice for entity behaving as a request broker (controller) in the system. Moreover, together with Pawel Kaczmarek, who developed the CMS skeleton of the system, I implemented user authorization and identification services inside of *Session Handling Agent*. Solution to this problem has been based on a concept of user session, popular in WWW server-side programming languages (PHP, ASP etc.). I also provided *Personal Agent* with profile exploitation algorithm to personalize recommendation. Eventually, mentioned problem of mixed initiative has been studied, in which both a user and Personal Agent are proactive actors in the system. Since Personal Agent can take over interaction with the user (due to limitations of client-server HTTP protocol and experiencing lack of control by the user in such a situation), push-and-pull technique has been proposed. In this solution, when a user request a system to perform an action on her behalf, a software agent (here *Personal Agent*) has the only chance to add to the system response questions whether user wants it to perform agent's actions.

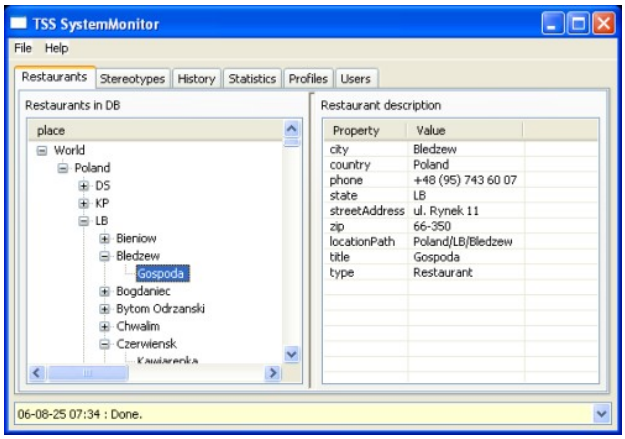


**Figure 1.** System Monitor shows presents data utilized and persisted in the system.

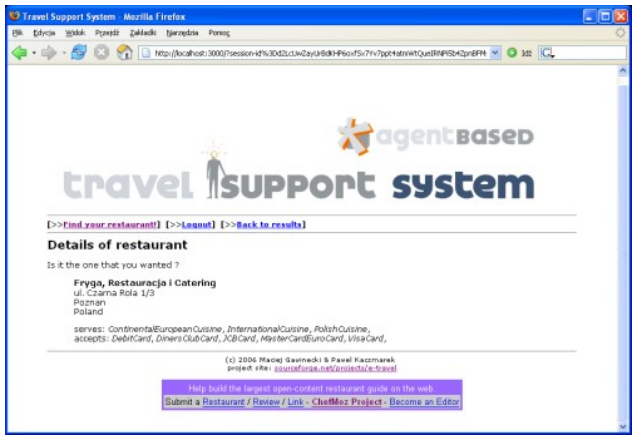


**Figure 2.** At the current stage the systems plays role of a restaurant recommender with ability to suggest any of 8700 Polish restaurants incorporated from Chefmoz dining guide. The user can access the system by any internet browser.

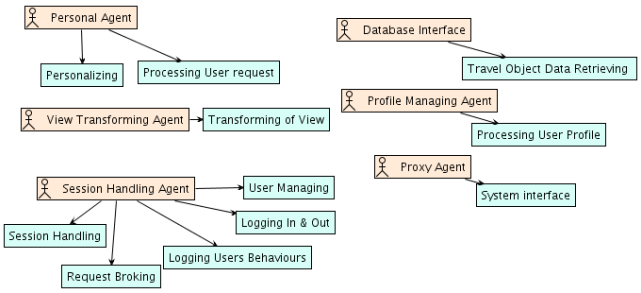


**Figure 3.** System agents and their roles. *Database Interface* is in fact realized by *Restaurant Service Agent*.

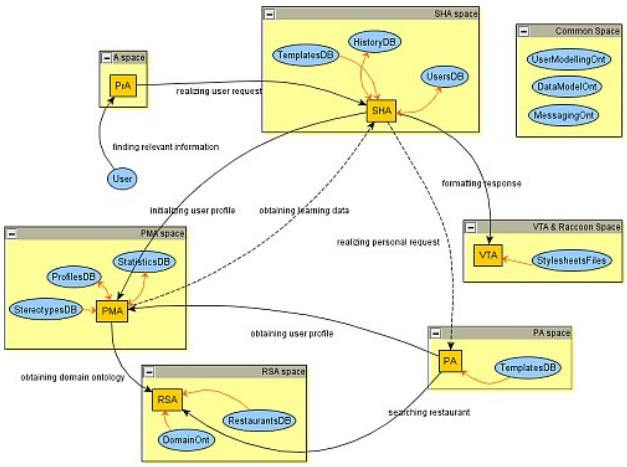


**Figure 4.** The diagram shows how agents depends on each other for their functionality.

## Obtained results

User modeling is a well-known technique for delivering personalized and relevant information to the user. In my work I discussed how it can be utilized in the context of the Semantic Web. More precisely, I proposed a complete process of constructing and management of user models (profiles) for an agent-based Travel Support System.

Since travel ontology is the centerpiece of the system it is naturally to represent user profiles as sets of statements about the "world of travel". This solution seems reasonable, allowing to preserve existing user profiles when the domain ontology is still in a development phase. Representing user profile as a set of opinions (statements) about domain is an innovative approach for domain demarcated as an ontology (in RDF). Such an approach allows, in the future, for propagating user opinions about known concepts from the domain into opinions about concepts, which are unknown, but related to the known ones. For example, if one is interested in castles and palaces, it could assumed, that she is generally interested in architecture. Relation between concepts is defined naturally in a domain ontology, and the whole process can be realized in terms of ontological reasoning.

Co-existence of two proactive entities in the system: the user (requesting a recommendation) and the *Personal Agent* (asking for opinion about recommended travel objects) results in the problem of mixed initiative. Since user comfort is crucial, an agent cannot break user-directed scenario by taking control over the dialog. Therefore, I proposed adaptation of push-and-pull technique into context of HTTP protocol.

I described also a novel algorithm for stereotyping to address problem of introducing new user to the system (the cold start problem). More precisely, the algorithm matches user with one of constructed stereotypes on the base of demographic data. Reasonable stereotypes must be developed for efficiency of stereotyping and thus I point the need of engaging psychologists and sociologists in stereotypes' development, since stereotypes people create are subjective and depend on the population context.

I have also noticed two sides of using feature-based techniques for filtering information from the Semantic Web. On the one side, adding a new object to the datastore will not cause cold start problem, if a new object is described by features, which have already got opinion in a user profile. On the contrary, partial lack of description of recommended objects (coming from Chefmoz service, Semantic Web in general) can negatively influence the user modeling process.

Finally, I presented practical application of the user modeling process in an travel support system, represented according to Prometheus methodology of agent-oriented programming. This application is available under GNU GPL licence at: http://e-travel.sourceforge.net/.

After defensing my Master Thesis, I have been still engaged in work on Travel Support System. I had the chance to look critically on some of proposed solutions and answer to considered

problems again. Particularly, I reconsidered the mentioned problem of how to distribute mentioned responsibilities to separate agents to find compromise between legibility of abstraction and communication performance in the work [7].

## References


1. Maciej Gawinecki, Zygmunt Vetulani, Minor Gordon, Marcin Paprzycki (2005) Representing Users in a Travel Support System. In: *Proceedings of the ISDA 2005 Conference*, IEEE Computer Society Press, Los Alamitos, CA, 393-398.
2. Maciej Gawinecki, Mateusz Kruszyk, Marcin Paprzycki (2005) Ontology-based Stereotyping in a Travel Support System. In: *Proceedings of the XXI Fall Meeting of Polish Information Processing Society*, PTI Press, 73-85.
3. Maciej Gawinecki, Minor Gordon, Ngoc Thanh Nguyen, Marcin Paprzycki, Michał Szymczak (2005) RDF Demarcated Resources in an Agent Based Travel Support System. In: M. Golinski et. al. (eds.), *Informatics and Effectiveness of Systems*, PTI Press, Katowice, 2005, 303-310.
4. Maciej Gawinecki, Minor Gordon, Marcin Paprzycki, Michał Szymczak, Zygmunt Vetulani, Jimmy Wright (2005) Enabling Semantic Referencing of Selected Travel Related Resources. In: W. Abramowicz (ed.) *Proceedings of the 8 th International Conference on Business Information Systems* (BIS 2005), Poznań University of Economics Press, Poznań, Poland, 271-288.
5. Maciej Gawinecki, Mateusz Kruszyk, Marcin Paprzycki, Maria Ganzha (2007) Pitfalls of agent system development on the basis of a Travel Support System. In: W. Abramowicz (ed), *Proceedings of the BIS 2007 Conference*, Springer, Berlin, LNCS 4439, 488-499.
6. McCauley, C., Stitt, C., Segal, M.: Stereotyping: from prejudice to prediction. Psychological Bulletin 87 (1980) 195–208
7. The Glossary of terms to the book Adaptable and Adaptive Hypermedia Systems, IRM Press, 2005. (http://www.dcs.bbk.ac.uk/~gmagoulas/publications.html)
8. Maciej Gawinecki, Mateusz Kruszyk, Marcin Paprzycki, Maria Ganzha (2007) Pitfalls of agent system development on the basis of a Travel Support System. In: W. Abramowicz (ed), Proceedings of the BIS 2007 Conference, Springer, Berlin, LNCS 4439, 488-499
9. Greer, J. i McCalla, G. 1993. Student modeling: the key to individualized knowledge-based instruction. NATO ASI Series F, 125.
10. Montaner, Miquel, López, Beatriz and de la Rosa, Josep Llu´is. 2003. A Taxonomy of Recommender Agents on the Internet. Artif. Intell. Rev., 19(4), 285–330.
11. Periklis, A. 2002. Data Clustering Techniques. Qualifying Oral Examination Paper, University of Toronto, Toronto, Kanada.
12. Sakagami, Hidekazu, Kamba, Tomonari i Koseki, Yoshiyuki. 1997. Learning Personal Preferences on Online newspaper articles for User Behaviors. Pages 291–300 z: Proceedings of 6th Int. World Wide Web Conference.
13. Pazzani, Michael i Billsus, Daniel. 1997. Learning and Revising User Profiles: The Identification of Interesting Web Sites. Mach. Learn., 27(3), 313–331.
14. Pawel Kaczmarek (AMU, 2005) Multimodal Communication Between Users and Software Agents.
15. Maciej Gawinecki, Minor Gordon, Paweł Kaczmarek, Marcin Paprzycki (2003) The Problem of Agent-Client Communication on the Internet. Scalable Computing: Practice and Experience , 6(1), 2005, 111-123
16. Fink, Josef i Kobsa, Alfred. 2002. User Modeling for Personalized City Tours. Artif. Intell. Rev., 18(1), 33–74.