# Representing Users in a Travel Support System

Maciej Gawinecki, Zygmunt Vetulani
*Faculty of Mathematics and Computer Science*
*Adam Mickiewicz University, Poznań, Poland*
*vetulani@amu.edu.pl*

Minor Gordon
*Department of Computer Science*
*Technical University of Berlin, Berlin, Germany*
*minorg@cs.okstate.edu*

Marcin Paprzycki
*Department of Computer Science*
*OSU, Tulsa, OK, USA*
*and*
*Computer Science, SWPS, Warsaw, Poland*
*marcin@cs.okstate.edu*

## Abstract

*We consider the construction and management of user profiles for an agent-based travel support system, with the goal of providing personalized content for individual users of the system. Profiles consist of statements about the "world of travel." Conditional probabilities for each statement model the strengths of user preferences. These probabilities are derived from implicit and explicit observations of user behavior.*

## 1. Introduction

The ability of sites such as Travelocity and Expedia to filter the mass of travel-related information on the Internet is key to their popularity. Content filtering and personalization is the antidote to information overload, and as such it plays an increasingly important role in our interactions with large-scale databases such as airline reservation systems and yellow page listings. Software agents have long been touted as a facilitating technology in this respect. The natural mapping between travel agents and software agents has inspired a number of attempts to design and implement an agent-based travel support system. Although most of these designs were purely conceptual, a few have proceeded to the implementation phase. We have developed an agent-based system for providing personalized travel-related content to Web users [2, 9, 13, 14, 24, 25]. In a recent paper [15] we have summarized current design of the system, which is to support needs of travelers by storing in a central repository semantically demarcated data (data gathering rather then indexing [12]). In this note we will discuss how the content personalization aspects of our design by focusing on how user profiles are created, managed and used. The most important contribution of this work is to go beyond the standard work on personalization, presented for instance in [8, 30, 31, 32, 33, 34, 36], where the exact form of data to be operated on is either ignored or is radically different from that which we have proposed.

We proceed as follows: in Section 2 we present the general architecture of the system and briefly sketch its functionality. Section 3 characterizes the travel-related data that the system manipulates and describes how this data is utilized in the context of user-system interactions (Section 4). In Section 5 we outline a basic user profile and methods for creating and updating profiles. Finally, in Section 6 we discuss how the user profile is actually employed in the system. We conclude the paper with description of current activities.

## 2. System architecture

The helicopter view of the architecture of the proposed system is depicted in Figure 1. Let us briefly summarize each of the components presented there (a more detailed description can be found in [15]).

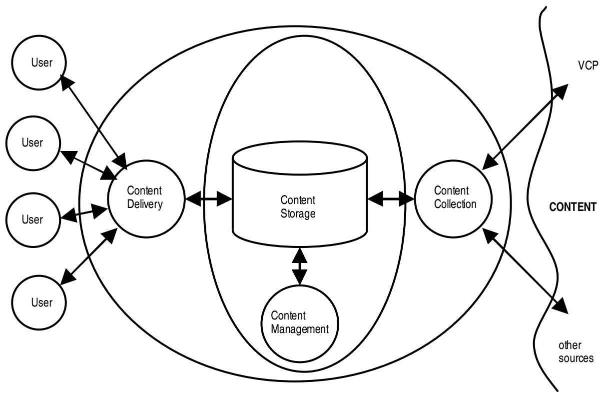


**Figure 1. Infrastructure for the travel support system**

*Verified Content Providers (VCP)*
Nowadays, a very large number of web sites contain travel-related information. However, due to the highly dynamical nature of the Internet, it is very difficult to successfully process available information [23]. Moreover, while the Semantic Web Project is to provide an answer to this situation [7], for the time being its main ideas remain "plans and wishes for the future." In reality, 99% of web content is based on HTML and other simple demarcations that are devoted to representing content in a human, rather than machine consumable format. Therefore, to address problems discussed in [23], and, more generally, to answer the question: "how can we provide users with the most accurate *and* most relevant information?" we utilize the concept of *Verified Content Providers* (*VCP*) [1].

A *VCP* is a site that is known to provide reliable and consistently available information (e.g. it does not "randomly" appear and disappear and/or change its format). It can be assumed that even though ontologically demarcated content is not available, it is possible to develop an interface for harvesting information and transforming it to such a form (see Content Collection Subsystem).

*Other Sources (OS)*
When dealing with *unverified* (vis-à-vis the *VCP*s), unstructured Internet-based information one has to consider: (a) its amount that makes an exhaustive search practically impossible, (b) unreliability of data, and (c) sources containing contradictory information that require application of data deconfliction techniques (e.g. fuzzy reasoning or rough set based techniques). Still, we should not discard such additional information and utilize it whenever possible. This approach will be even more important when the Semantic Web [7] will become popular among small independent sites making data available there much easier to process automatically.

*Content Collection Subsystem* (*CCS*)
In the proposed approach, the information gathered from the *VCP*s and *OS* is stored in a form of semantically demarcated "tokens" describing travel resources. We have selected the RDF as the ontology tagging "language" [28]. Making this decision, JENA [18] became an immediate choice for the technology for storing RFD triples. Thus, in the *CCS,* sets of RDF triples defining travel object (according to the system-defined ontology [11]) collected from the Internet (and possibly other repositories) are stored in the, JENA-based, central repository. Note that questions related to the size and to the centralized nature of the repository are outside of the scope of this note.

*Content Management Subsystem* (*CMS*)
This subsystem involves all functions related to management of data stored in the central repository. Here, we are particularly interested in incomplete tokes delivered by the *CCS* (e.g. restaurant that is missing phone number, or hotel that is missing information about amenities) and tokens that contain time sensitive information (e.g. programs of movie theaters that change every Friday). Obviously, even information that is not explicitly time-sensitive (e.g. Museum opening times) has to be re-verified in some unspecified time intervals. The *CMS* provides infrastructure that is to assure reliability of the information stored in the system.

*Content Delivery Subsystem* (*CDS*)
This subsystem is the one that we are particularly interested in. Here the data stored in the central repository is manipulated for the delivery to the user. Agents in this subsystem receive a query from the user and work to provide information responding to that query in a way that is matching her personal preferences. We discuss this process in more detail in subsequent sections.

*Users*
The system is to be accessible to any Internet-enabled device, ranging from standard PC-based browsers to palmtops and WAP-conversant phones etc., and even non-human entities (such as other agents). Upon completion, system responses are delivered to the user in the format that matches specific requirements of the device (see [10, 16] for more details).

## 3. Data in the system

Travel-related data for our travel support system is stored in form of semantically demarcated travel object tokens (sets of RDF triples). Each complete token represents an instance of a concept defined by our travel ontology. Thus far we have developed and "implemented" complete ontologies of a hotel and a restaurant (see [11] for examples and a discussion of how these ontologies are related). Figure 2 shows a fragment of the restaurant ontology. Note the cuisine "branch" and the "payment method" branch.

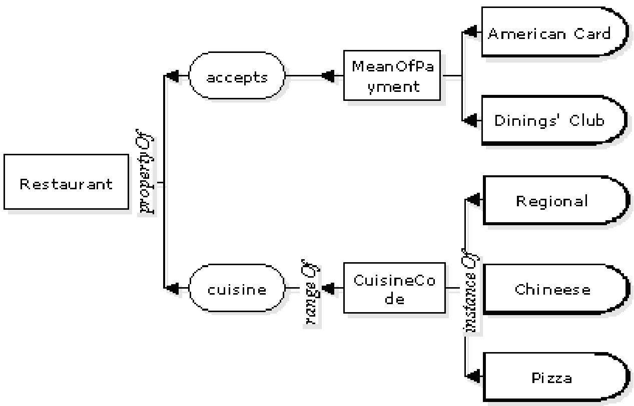


**Figure 2. Fragment of restaurant ontology**

For each travel object known to the system we store a complete set of statements for all properties of the ontology definition. For instance, a restaurant might be described by set of RDF statements (in N3 syntax) shown in Figure 2:

```
@prefix :
 <http://www.agentlab.com/db/chefmoz.rdf#>.
@prefix money:
 <http://www.agentlab.com/schemas/money.rdf#>.
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix res:
 <http://www.agentlab.com/schemas/restaurant.rdf#>.

<#Poland_ZP_Swinoujscie_Victoria__Kawiarnia1050440379>
 a res:Restaurant
 ; res:accepts
          money:AmericanExpressCard,
          money:DebitCard,
          money:DinersClubCard,
          money:JBCCard,
          money:MasterCardEurocard,
          money:VisaCard
 ; res:cuisine res:CafeCoffeeShopCuisine.
```

Let us now describe how the RDF demarcated data is "used" in the system.

## 4. Content Delivery Subsystem

The Content Delivery Subsystem (*CDS)* of our agent-based travel support system responds to user queries by delivering personalized content. In our earlier work we have shown that user-agent communication is a non-trivial task, and presented a solution to this problem [16]. Let us therefore assume that it is possible to submit a query to the system from any Internet-enabled device. Such a query, formulated, for instance, by filling out an appropriate form is transmitted to the *Proxy Agent* (*PrA*) residing on a "gateway server." Regardless of the form of the query the "query content" is extracted by the *PrA*, wrapped into an ACL message and sent to the *Personal Agent* (*PA*). The *PA* forwards the message in two directions. First, to an agent that will store the message contents in a *user behavior database* (where all user queries sent to the system and all system responses are logged for future processing – see also [8]: Section 5 and Figure 3); and second, to the *Database Agent* (*DBA*). The *DBA* acts as an interface to our database of RDF statements. It translates the query into the RDF Query Language (RDQL) language and executes the query. As a result, a collection of object tokens (sets of RDF triples) is returned. These RDF triples (tokens) are then sent to the personalization infrastructure (Figure 3), which consists of a number of *RDF Agents* (*RDFA*). Each *RDF Agent* is responsible for applying one of more simple rules to the result set, e.g. by issuing further queries and expanding the result set. Rules are of the type "Sichuan food is also Chinese food" or "Romantic Comedy is also a Comedy." The *RDFA*s operate as a team, passing the result set, wrapped in ACL messages, from one to the next. Their role is to maximize the set of responses to be delivered to the user (at this stage no potential response is removed form the set). While the work of *RDFA*s is important for content delivery, due to lack of space, we focus our attention on user profiles, their creation, management and utilization.

The augmented set of tokens produced by the team of *RDFA*s is sent back to the *PA*. The *PA* utilizes the *user profile* to filter (personalize) the response. For instance, the *DB* and the *RDFA*s may not know that the user never stays in Howard Johnson hotels and never goes to Braum's restaurants. Travel objects representing these two chains may be included in the preliminary set of travel objects, but later the *PA* will eliminate them from the set returned to the user.

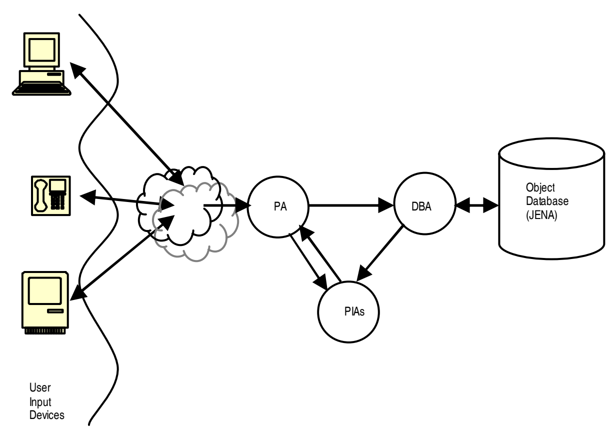


**Figure 3. Content delivery subsystem: PA – personal agent, DBA – database agent, PIA – personalization infrastructure agents, "the cloud" – the user device ↔ agent system interface (among others Proxy agent and Racoon).**

The answer set is send to the Racoon server [27] to render the response to be displayed on user device as well as to the personalization infrastructure to be logged (in the same *user behavior database* as the query was logged in) to form query-response pairs that represent user-system interactions. These pairs can then be utilized to modify individual user profiles as well as to develop group profiles (stereotypes) as well as study trends in user behavior [8, 22].

## 5. Creation and management of user profiles

User profiles in our system are primarily based on models presented in [6, 8, 19, 21]. We believe that the most natural way of building user profile is to utilize a domain ontology (in our case the travel ontology) underlying our system. Furthermore, the resulting profile must be incrementally adjustable to keep up with changing interests and preferences. Proposed structure should also be universal enough to allow us to create and store "stereotypes" representing our current knowledge of preferences of users belonging to various groups positioned at different levels of social stratification [22]. Let us look now at various aspects of user profile representation, creation and management.

*User Profile Representation*

A user profile is composed of statements indicating a user's opinion of objects from the domain ontology (see Figure 2 and subsequent paragraphs). We use RDF reification to attach meta-statements to the opinion statements [28], as in: *Chinese cuisine is user's favorite with probability X*. With these meta-statements of probability we can build a probability graph analogous to a Bayesian network [4, 34]. Further conditional probabilities can be derived from relations in the domain ontology: *subclassOf*, *propertyOf*, *instanceOf* and *rangeOf*. In other words, if a statement expresses a user's opinion about a domain concept the meta-information (through appropriate manipulation of probabilities) can be extended via these relationships to related concepts. Observe that the same profile structure can be used to represent individual users as well as abstract user-group stereotypes.

*Initial Profile*

Collecting new user preferences is one of the primary challenges of a profiling system [8]. Our proposed solution is to employ stereotyping – creating an initial user-group-model from predictions about the user based on some classification [19, 22]. Our user model is initiated by fitting users to stereotypical descriptions [29], representing the features of classes of users. Initially, these stereotypes will be acquired from responses to a survey conducted among potential users of the system. Actual users of the system will also be asked to fill-in the questionnaire (gathering, for instance, demographic and domain-based data) and their responses will be matched against the existing stereotypes. As the system operates we will be able to adapt individual user profiles as well stereotypes by observing individual interest patterns (see below) and applying data mining techniques to the data logged in the *user behavior database* to reveal group behavior [8, 22].

*Relevance Feedback*

The recommender agent (*PA*) needs relevant data to update the user profile over time. To achieve this goal, for each user we collect a complete log of interactions with the system. This log represents both positive ("she selected the *Y* restaurant") and negative ("he never selected the *Z* hotel") *implicit feedback* [21, 34]. We will also solicit *explicit feedback* from users in the form of rating suggestions. Information gathered through both methods will be used to adjust the probabilities attached to statements in the user's profile.

Furthermore, we can derive changes to the user profile in the form of associations. For example, a user's explicit or implicit behavior in relation to a particular restaurant can interpreted as an opinion on them more general characteristics of the restaurant, like "Chinese cuisine" or "separate section for smokers." Preference for one concept over another can be inferred from the frequency with which a given feature is represented in the user's history (*favouriteProb*) as well as domain associations (*fromDomainProb*). For the purpose of our system we adapt history-based learning proposed in [6]. The learning process consists of the following stages:

(1) compute frequency of occurrence of different concepts in the user's history (relatively to the history of all users) – this must be done separately for the *implicit* and the *explicit* feedback
(2) compute the *favouriteProb* by combining results obtained in step (1), while more weight is given to the *explicit* feedback
(3) for each node in the profile perform the following two steps
    a. compute the *fromDomainProb* values by performing domain interference – this is achieved using upwards propagation of probabilities (the *favouriteProb* is propagated from leaves (nodes) to super-concepts in of the probability graph [6])
    b. combine the *favouriteProb* and the *fromDomainProb* into the *interestingProb*;
(4) results derived in step (3b) are then used to distinguish concepts in the user's profile that are *significantly interesting*, *significantly uninteresting* or *unclassified.* These classifications result from application of a standard univariate significance analysis [6], i.e. if a feature appears in the user's history less frequently than in a random sample, we

can consider the user not interested in the concept, and vice versa.

## 6. Application of user profile

Let us now briefly discuss how *user profile* can be used in our system.

*Utilization of the Profile*

Without loss of generality we will use restaurants as example of profile utilization. After the *RDFA*s complete their work, a “maximum” set of travel objects is delivered to the *PA*, which is asked to filter and order the objects according to the user’s preferences represented by the user profile (we are thus dealing with feature-based recommending). We compute the probability that a given restaurant is the user's favorite by: (1) cropping the sub-graph consisting of paths between the *Restaurant* concept and the features of the restaurant that are *significantly interesting* for the user, then (2) computing the total probability of the resulting sub-graph. Furthermore, the value of *interestingProb* is increased for each concept appearing in the current query thus strengthening its direct influence on the resulting recommendation (it is assumed that the fact that a given feature is directly specified in the query indicates its current importance to the user). Finally, *interestingProb* is also increased if the restaurant was some time in the past explicitly rated by the user as interesting (results of such process are presented in Figure 4). The resulting probabilities are used to filter and rank restaurants. Restaurants that are ranked as *significantly uninteresting* will have probability close to 0 and thus can be removed from the set. The remaining restaurants with probabilities above a certain threshold will be ranked accordingly (starting from those with the highest probability that will be displayed first).

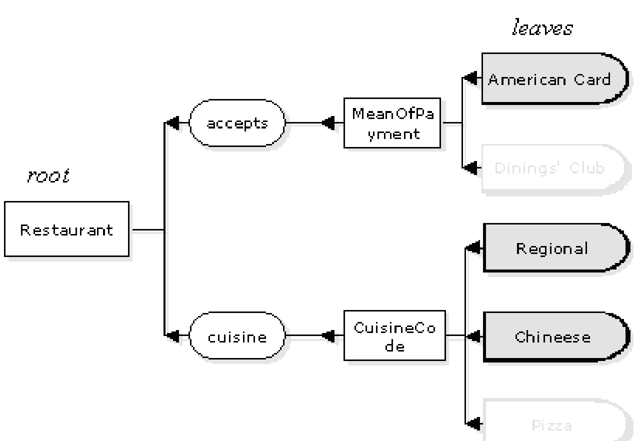


**Figure 4. Sub-graph of the probability graph based on the restaurant serving: Regional and Chinese cuisines and accepting American Express. Transparent leaves were found not significantly interesting or do not exist.**

*Combining with other recommending techniques*

The personalization infrastructure proposed here may also employ different recommending techniques in parallel and combine the results. For example, one agent can interact with other agents for collaborative filtering, while another agent makes feature-based recommendations. Various recommendations can be combined utilizing a weighted average [3].

## 7. Concluding remarks

In this note we have outlined the high-level architecture for an agent-based travel support system and introduced content personalization into the system design. We have described how user profiles can be represented in a system that is based on RDF demarcated data. The proposed user profile incorporates the concepts of the ontology developed for the system, allowing us to consult the profile for content filtering and ranking.

We are currently in the process of implementing most of the key parts of the system. We are using JADE agent environment, JENA repository for the RDF demarcated data, and Racoon for rendering responses for variety of devices. We are in the process of implementing the above described content personalization framework (with the central repository and the content delivery skeleton already fully operational [15]). We will report on progress of implementation in the near future.

## References

[1] Abramowicz W., Kalczyński P. J. (2002) “Building and Taking Advantages of the Digital Library for the Organizational Data Warehouse,” in: Cobb M. et. al. (eds.), Proceedings of the Second Southern Conference on Computing, Hattiesburg, Mississippi, USA, October 26-28, 2000, to appear

[2] Angryk R., Galant V., Gordon M., Paprzycki M., “Travel Support System – an Agent Based Framework,” in: H. R. Arabnia, Y. Mun (eds.), Proceedings of the International Conference on Internet Computing (IC’02), CSREA Press, Las Vegas, NV, 2002, 719-725

[3] Burke R., "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, 12 , 4 , 331 - 370

[4] Ding Z. Peng Y., "A Probabilistic Extension to Ontology Language OWL," Proceedings of the 37th Hawaii International Conference on System Sciences - 2004

[5] ebXML R/R: http://ebxmlrr.sourceforge.net/

[6] Fink J., Kobsa A., "User Modeling for Personalized City Tours," Artificial Intelligence Review, 18, 2002, 33–74

[7] Fensel D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer, Berlin, 2001

[8] Galant V. and Paprzycki M. (2002) “Information Personalization in an Internet Based Travel Support System,” in:

W. Abramowicz, Proceedings of the BIS’2002 Conference, Poznań University of Economics Press, Poznań, Poland, 2002, 191-202

[9] Galant V., Gordon M., Paprzycki M., “Knowledge Management in an Internet Travel Support System,” in: B. Wiszniewski (ed.), Proceedings of ECON2002, ACTEN, Wejcherowo, 2002, 97-104

[10] Galant V., Gordon M., Paprzycki M., “Agent-Client Interaction in a Web-based E-commerce System,” in: D. Grigoras (ed.), Proceedings of the International Symposium on Parallel and Distributed Computing, University of Iaşi Press, Iaşi, Romania, 2002, 1-10

[11] Gawinecki M., Gordon M., Paprzycki M., Szymczak M., Vetulani Z., Wright J., “Enabling Semantic Referencing of Selected Travel Related Resources,” in: W. Abramowicz, Proceedings of the BIS’2005 Conference, Poznań University of Economics Press, Poznań, Poland, 2005, 271-290

[12] Gilbert A., Gordon M., Nauli A., Paprzycki M., Williams S., Wright J., “Indexing Agent for Data Gathering in an e-Travel System,” Informatica, Vol. 28, No. 1, 2004, 69-78

[13] Gilbert A., Gordon M., Paprzycki M., Wright J., “The World of Travel: a Comparative Analysis of Classification Methods,” Annales UMCS Informatica, A1, 2003, 259-270

[14] Gordon M., Gilbert A., Paprzycki M., “Knowledge Representation in the Agent-Based Travel Support System,” in: T. Yakhno (ed.) Advances in Information Systems, Springer-Verlag, Berlin, 2002, 232-241

[15] Gordon M., Paprzycki M., “Designing Agent Based Travel Support System,” Proceedings of the ISPDC Conference (to appear)

[16] Gordon M., Paprzycki M., Gawinecki M., Kaczmarek P., “The Problem of Agent-Client Communication on the Internet,” PDCP, to appear.

[17] Harrington P., Gordon M., Nauli A., Paprzycki M., Williams S., Wright J., “Using Software Agents to Index Data in an E-Travel System,” in: N. Callaos (ed.), Electronic Proceedings of the 7th SCI Conference, Orlando, 2003, CD, file: 001428.pdf, 6 pages

[18] JENA: http://www.hpl.hp.com/semweb/

[19] Kobsa A., Koenemann J., Pohl W., „Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships,” The Knowledge Engineering Review, 16:2, 2001, 111-155

[20] Maes P., “Agents that Reduce Work and Information Overload,” Communications of the ACM, 37(7), 1994, 31-40

[21] Montaner M., López B., De La Rosa J. L., "A Taxonomy of Recommender Agents on the Internet," Artificial Intelligence Review, 19, 2003, 285–330

[22] Nistor C. E., Oprea R., Paprzycki M., Parakh G., "The Role of a Psychologist in E-commerce Personalization," Proceedings of the 3rd European E-COMM-LINE 2002 Conference, Bucharest, Romania, 2002, 227-231

[23] Nwana H., Ndumu D. (1999) “A perspective on software agents research,” The Knowledge Engineering Review, 14 (2), pp. 1-18

[24] Paprzycki M., Angryk R., Kołodziej K., Fiedorowicz I., Cobb M., Ali D. and Rahimi S. (2001) “Development of a Travel Support System Based on Intelligent Agent Technology,” in: S. Niwiński (ed.), Proceedings of the PIONIER 2001 Conference, Technical University of Poznań Press, Poznań, Poland, pp. 243-255

[25] Paprzycki M., Kalczyński P. J., Fiedorowicz I., Abramowicz W. and Cobb M. (2001) “Personalized Traveler Information System,” in: Kubiak B. F. and Korowicki A. (eds.), Proceedings of the 5th International Conference Human-Computer Interaction, Akwila Press, Gdańsk, Poland, pp. 445-456

[26] Poslad S., Laamanen H., Malaka R., Nick A., Buckle P., Zipf A. (2001) CRUMPET: Creation of User-friendly Mobile Services Personalised for Tourism. Proceedings of: 3G 2001 - Second International Conference on 3G Mobile Communication Technologies. 26-29 March 2001. London, UK. http://conferences.iee.org.uk/3G2001/

[27] Racoon: http://rx4rdf.liminalzone.org/Racoon

[28] RDF Primer, http://www.w3.org/TR/rdf-primer

[29] Rich E., "User Modeling via Stereotypes" in "Cognitive Science 3", 1979, pp. 329-354

[30] Rud O. P. (2001) “Data Mining Cookbook. Modeling Data for Marketing, Risk, and Customer Relationship Management,” Wiley, New York, NY

[31] Simon A. R., Shaffer S. L. (2001) “Data Warehousing and Business Intelligence for E-commerce,” Morgan Kaufman, New York, NY

[32] Soltysiak S., Crabtree B. (1998) “Automatic learning of user profiles – towards the personalization of agent service,” BT Technological Journal, 16 (3), pp. 110-117

[33] Suarez J. N., O’Sullivan D., Brouchoud H., Cros P. (1999) “Personal Travel Market: Real-Life Application of the FIPA Standards.” Technical Report, BT, Project AC317

[34] Wittig F., "ML4UM for Bayesian Network User Model Acquisition" MLIRUM'03: Second Workshop on Machine Learning, Information Retrieval and User Modeling

[35] Wright J., Gordon M., Paprzycki M., Williams S., Harrington P., “Using the ebXML Registry Repository to Manage Information in an Internet Travel Support System,” in: W. Abramowicz and G. Klein (eds.), Proceedings of the BIS'2003 Conference, Poznań University of Economics Press, Poznań, Poland, 2003, 81-89

[36] Zhang B., Li W., Xu Z. (2002) “Personalized Tour Planning System Based on User Interest Analysis,” Proceedings of the BIS’2002 Conference, Poznań, Poland, April, 2002, pp. 184-190