Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide

Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide

Federica Cena [a,b], Luca Console [a], Cristina Gena [a], Anna Goy [a] , Guido Levi [b], Sonia Modeo [a,b] and Ilaria Torre [a]

[a] *Dip. di Informatica, Universita' di Torino, Torino, Italy*
E-mail:
{*cena,lconsole,gena,goy,modeo,torre*}*@di.unito.it*
[b] *CSP Innovazione nelle ICT s.c.a.r.l., Torino, Italy*
E-mail: levi@csp.it

Intelligent adaptation is a key issue for the design of flexible support systems for mobile users. In this paper we present UbiquiTO, a tourist guide which integrates different forms of adaptation: (i) to the device being used (web access via Laptop, PDA, smartphone), (ii) to the user and her features and preferences (personalized interaction), (iii) to the context of interaction, and in particular to the user location, besides some other environment features such as the time of the day. UbiquiTO adapts the content of the service being provided (recommendation and amount/type of information/features associated with each recommendation) and the presentation (interface). In order to achieve better performance it keeps track of the user behavior, updating and refining the user model during the interaction. In the paper we introduce the architecture of the system and the choices we made as regards user, device and context modeling and adaptation strategies. We also present the results of a preliminary evaluation of the system behavior.

Keywords: Adaptive systems, User modeling, Context modeling, Ubiquitous information access

1. Introduction

Mobile access to information and services is becoming more and more important, especially for people who travel frequently for work. The possibility of accessing the same services using different devices, in different locations and in different contextual conditions would be very important for such mobile workers. However, the design of systems that can effectively fulfil such a need is a challenging problem for research.

Several issues have to be faced. Mobile access to services is very peculiar as in most of the cases the user has no chance to browse a set of alternatives. This difficulty may be due to the limitations of the device being used (browsing is difficult on a smartphone or PDA whose screen allows to present only a few pieces of information at a time) or to the connectivity constraints (connection via mobile devices is expensive, bandwidth is low and not always reliable) or to the fact that the user is often in a hurry and needs some support immediately. Moreover, mobile equipments are different from standard laptops or desktops in terms of I/O devices they support and obviously in terms of usage. Finally continuity should be supported, in the sense that a user may access the same services in different contexts (e.g. using different devices at different time) and no abrupt changes must be felt when moving from a context to another.

This calls for the design of intelligent systems supporting ubiquitous access to services and providing the user with the desired service or information, in the right way, at any time and location, and using any device. The ultimate goal is to design intelligent assistants supporting the user in a smart and possibly cooperative way in her access to services.

Adaptivity has been recognized as one of the main handles to achieve such a goal [5], [38], [8]. Indeed, different forms of adaptation can be adopted. On the one hand, the service may *adapt to* different aspects such as the device being used (e.g., PDA vs smartphone vs laptop), to the user and

her peculiar needs and preferences ("personalization" of the services), to the current user location ("location-based" services), or even to other aspects of the context of use (e.g., the fact that the user is moving or not, the fact that she is driving a car, the time of the day ...).

On the other hand, different features of the service could be adapted (*adaptation of*), ranging from the type of service (*content adaptation*), to the way it is presented or accessed (*interface adaptation*), to the modality of interaction (e.g., "push" vs "pull" services). Combining all these aspects would allow for a very effective and flexible service provision.

In this paper we focus on tourist services and we present the architecture of UbiquiTO, an agent-based system that acts as an expert tourist guide for mobile users, integrating the forms of adaptation mentioned above.

In Section 2 we introduce the main goals of the project by providing a short usage scenario. In Section 3 we describe the architecture of the system, in Section 4 the initialization and update of the user model; in Section 5 we present some details about the different forms of adaptation. Finally, Section 6 describes the results of a test performed to evaluate the system behavior, Section 7 presents some related works and Section 8 outlines some implementation issues. Section 9 concludes the paper.

2. Goals and Usage Scenario

The UbiquiTO project has been developed at the Computer Science Department of the University of Turin, in collaboration with the CSP Research Center (http://www.csp.it/eng/), in order to support the local government in the improvement of tourist services, which will be provided during the 2006 Olympic Winter Games. The current prototype provides tourist information about the city of Turin.

In order to introduce the goals of UbiquiTO, let us start with the sketch of a potential usage scenario. Let us suppose that John is a manager, middle aged, mobile worker with interest in gastronomy. He has been invited to a corporate conference in Turin. On the conference web site he finds the link to access the UbiquiTO service and he signs up (he might also register at the Turin Airport Information Desk or in the hotel). At the end of the conference, at five o'clock, he receives on his PDA an advice to visit the National Museum of Cinema (which is close to the conference site) and to try the Italian "aperitivo" in the indoor bar. The system also proposes a typical restaurant close to John's hotel, by inferring from his profile that he likes tasting local food. However, John needs to move to another hotel to meet some colleagues. He provides the system with the meeting location and so UbiquiTO suggests him another restaurant, serving the same type of food but closer to John's new location.

The sketched scenario allows us to highlight the main goals of the project. The application aims at providing tourist services to heterogeneous types of visitors, although the first prototype focuses on mobile workers. The main reason for this choice is that in a short-medium term scenario, we foresee that business travellers can be people with the highest availability of PDAs, advanced mobile phones, and GPS equipped devices. Moreover, they are probably less worried about connection costs and have higher probability of recurring visits, thus allowing the system to collect information for building a stable and reliable user profile.

As mentioned in the introduction, the success in the provision of mobile services depends on the possibility to adapt such services to the usage context and to the user features. UbiquiTO integrates different types of adaptation:

- *User*: the system maintains a profile of the user, including her interests, preferences and the history of her previous visits to Turin, and exploits this profile in order to tailor its suggestions to the user preferences. The user profile can be updated either explicitly, by the user herself (a "modify your profile" link is available), or by means of automatic learning mechanisms.
- *Device*: the user interface adapts to different types of devices. The prototype is focused on personal computers, PDAs and smartphones, and we are working for extending the system to other devices, such as vehicle on-board navigation systems, and DTT (Digital Terrestrial Television).
- *Context*: tourist services are provided according to a location-based strategy, i.e., they depend on the specific area (and coordinates) of the town where the user is (or is expected to

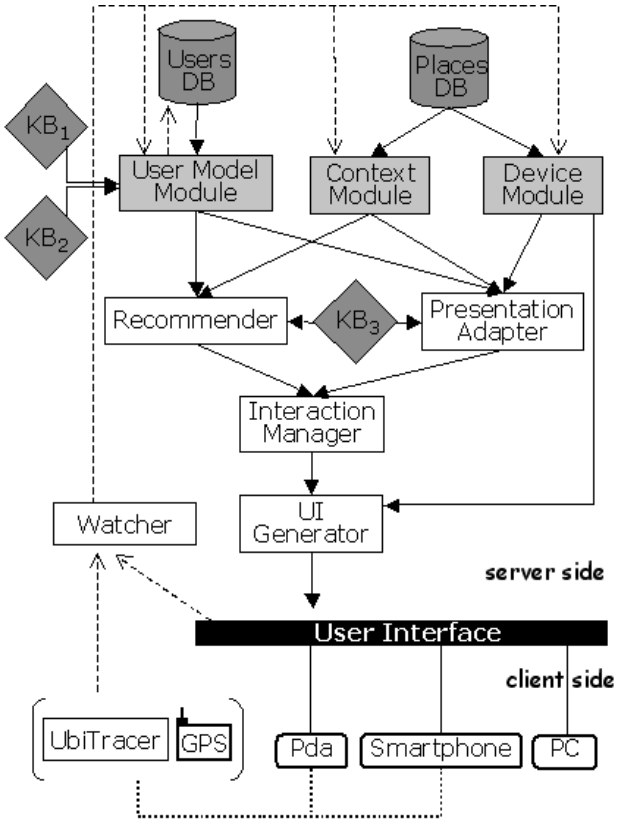

Fig. 1. The architecture of the prototype.

be) at the time the service is requested. The system considers also a set of further parameters like the time of the day, or the fact that the user is moving, and adapts the interaction taking them into account.

3. The Architecture

The architecture of the prototype is reported in Figure 1.

The main agents are located on the server, in order to support flexibility and independence of device.

On the server side, the system is composed of two *personalization* agents (the Recommender and the Presentation Adapter) and three *support modules* (the Interaction Manager, the UI Generator and the Watcher). They interact with a set of specialized *modeling modules* (User Model, Context, and Device modules) in order to get the information needed to adapt service provision, content and presentation to the user features, the device she is using and the context.

On the client side, the system includes a location agent (UbiTracer), which may be installed on GPS-equipped PDAs and smartphones.

In the following we will describe the agents functionalities and the role of the modules. After that, in Section 5, we will discuss in more detail the different adaptation strategies implemented by the system.

The Recommender and the Presentation Adapter agents represent the core of the system. They are in charge of the personalization. In particular:

- The *Recommender* handles the adaptation of the content. It exploits personalization rules to suggest items (e.g. places to visit, accommodations, restaurants, etc.) tailored to the user preferences and to the context (mainly, location).
- The *Presentation Adapter* exploits adaptation rules to adapt the presentation (e.g. the amount of information to be displayed and its layout) to the user preferences, to the device characteristics and to the context.

These agents are supported by other modules which carry on the interaction with the user.

- The *Interaction Manager* handles the dialog with the user; each dialog step corresponds to the generation of an XML object representing the personalized content of the page to be displayed.
- The *UI Generator* handles the application of XSL stylesheets that transform the XML objects into the (X)HTML pages representing the User Interface (UI), taking into account the different characteristics of the devices and the context features.
- The *Watcher* collects the inputs provided by the user, implicitly while interacting with the system (e.g. GPS coordinates, page views, etc.), or explicitly (e.g. registration form data). Given the diversity of inputs, the Watcher is structured in a set of different subcomponents, each one collecting and storing a specific type of data (e.g. as it will be detailed in Sections 5.1.1, the ubiStorer gathers position/time data, etc.). These data are exploited by the User Model module, the Context module and the Device module (see Figure 1) in order to produce/update the corresponding models.

- The *UbiTracer*, on the client side, is a location agent which can be downloaded on PDAs and smartphones. It acquires the spatial coordinates of the user from the GPS receiver and sends them to the Watcher on the server. Notice that non GPS-equipped devices may also use the system, providing the current location of the user by clicking on a web map displayed by the system.

As mentioned, the architecture also includes three specialized modeling modules:

- The *User Model module* handles the models (profiles) of users. It initializes and revises/updates the models. Both tasks are accomplished using knowledge that relates data (facts) collected by the Watcher to user features.
- The *Context module* determines the current position of the user, analyzing data collected by the Watcher and provides it to the Recommender. Furthermore, it manages some other information, such as the fact that the user is moving or not, that are used to adapt the presentation to the environment conditions.
- The *Device module* exploits the parameters of the HTTP dialog with the client forwarded by the Watcher to find out the type of device the user is connecting with and its features.

3.1. The Knowledge Base

UbiquiTO knowledge is composed of three knowledge bases (KB) and two databases (DB).

Knowledge Bases.

Two KB are used by the User Model Module and another KB is exploited by the Recommender and Presentation Adapter. The first User Modeling Knowledge Base (KB_1 in the figure) contains a set of rules for the initialization of the user model starting from data explicitly provided by the user. The second Knowledge Base (KB_2) is composed of another set of rules for the update of the user model, starting from user's events collected by the system. Finally, the Adaptation Knowledge Base (KB_3) is exploited by the Recommender and Presentation Adapter to personalize the interaction. These knowledge bases will be explained respectively in Sections 4.1, 4.2, and 5.

Databases.

In UbiquiTO there are a Users DB and a Places DB.

Users DB. This database stores data used to define the user models, plus auxiliary information required for managing user sessions. As it will be explained in the next section, two types of user modeling data are used: features directly elicited from the user, which include socio-demographic data, interests categories and data related to the visit (duration, free time); inferred features which regard interests in sub-categories (specific interests) and propensity to spend. Notice that user interests are defined in a two-level hierarchical structure. Top level categories include: food, music, cinema, art and shopping; the corresponding sub-categories are traditional/ethnic/fast/etc. for food, classical/rock/pop for music, etc. Indeed, the degree of interest for categories is explicitly required to the user, while the one for sub-categories is inferred.

Places DB. This database stores all the items that can be presented to the user, namely all the tourist places (including tourist attractions, as well as hotels, restaurants, and so on) that can be suggested. Given the goals of adaptation (see Sections 5.1, 5.2), this database contains information that allow to recommend items on the basis of the current user position and her preferences and to generate different presentations of each item, varying the detail level and the amount of information. Therefore, for each item, the database stores: i) spatial information, for locating the place; ii) classifying information, for enabling the match with user preferences; iii) descriptive information, for generating different versions of item descriptions. The first two types of information are also used for updating the User Model. In fact, from the features of the items selected by the user, we can hypothesize information about user's preferences (see section 4.2). In more detail:

i) Spatial information concerns latitude, longitude and zone. The first two data are stored as GPS coordinates, the zone attribute is a code used for non-automatic localization (see Section 5.1.1).
ii) Classifying information concerns a set of features that are different for different classes of items. In particular, places are split in four main classes: events (e.g. theatre, concerts, exhibitions, etc.), places to visit (his-

torical buildings, museums, etc.), places for eating and drinking (restaurants, pub, etc.) and accommodations (hotels, bed and breakfast, etc.). Each class is characterized by a set of features. For example, for the events class we store features such as category (e.g. theatre, cinema, etc.), genre (cabaret, opera, etc.), time of the event, cost (maximum and minimum price and qualitative value - high, medium, low), discounts. For places for eating and drinking we consider: type of food served (e.g. traditional, ethnic, etc.), menu (main dishes), category (e.g. restaurant, pub, etc.), style (elegant, country, etc.), cost and closing day. These features are taken into account by the Recommender, which evaluates their match with user features. For instance, cost features are evaluated in relation to user propensity to spend; event category and genre features to user interests; places for eating and drinking category to age and propensity to spend, etc. Time of the event and closing day are used for excluding items that do not match with the user time availability.

iii) Descriptive information concerns the storage of four different descriptions for each item. These descriptions are the result of the combination of two pairs of data: long vs short description, and detailed vs essential description. The first pair is used for adapting the amount of displayed information, on the basis of the current device (quantitative difference between descriptions), the second one is used for adapting the detail level in accordance with the user interest (qualitative difference between descriptions).

As it can be noticed, the stored information represent a compromise between the need of conforming our DB structure to the one of common tourist information providers and the need to add extra information for increasing the possibilities of match with the user features and thus improving adaptation.

4. User Modeling

As sketched in the previous section, UbiquiTO adapts the recommendations and their presentation on the basis of the User Model (UM), the device and the context. The knowledge for building up the UM knowledge bases was elicited from three different sources:

1. The knowledge used in MastroCARONTE [14], an on-vehicle tourist system that suggests the same categories of items provided by UbiquiTO. MastroCARONTE knowledge was based on a survey about the lifestyle of the Italian population. Given the good results exibited by the system [18], we decided to reuse part of its knowledge base in designining UbiquiTO.
2. A survey about tourism in Turin that specifies profiles, behavior and preferences of foreign tourists [9].
3. A questionnaire we personally gave out to tourists in strategic places of the city of Turin, such as the central train station, the airport, etc.

4.1. The Current User Model

The Current User Model (UM_C) represents a picture of what the system knows about a specific user. It is structured in two parts:

1. *Explicit data:* socio-demographic data (age, gender, profession), general interests (music, food, cinema, art), data related to the visit (duration, free time).
2. *Inferred data:* propensity to spend and specific interests (classical/rock/pop music, traditional food, fast food, etc).

In part 1) data are represented as feature-value pairs, where the values are those selected by the user during the registration process.

While socio-demographic data and data related to the visit are represented as alphanumeric values (e.g., age: 45; gender: male; profession: manager, etc), general interests have a score, in the range $0 \ldots 1$, that is quantitative indicator in which 1 represents the maximum interest and 0 denotes "no interest" (e.g., interest in food: 0.5, interest in music: 1, etc).

The Current UM is initialized with this explicit information provided by the user, in order to avoid the well known cold start problem [36] of recommender systems.

Part 2) represents psychographic data and specific interests, which are inferred by the systems using rules having the following format:

IF (logical combination of feature-value pairs)
THEN (inferred user features-value pairs)

and are represented as feature-value pairs (e.g., shopping interest: 0.3). For example, a rule which could fire for the user John, described in Section 2, is:

IF ((age >= 45)
AND (profession = manager))
THEN PropensityToSpend = 0.7

The values are quantitative *indicators*, ranging in the interval [0..1] again. In order to define more reliable rules, personalization strategies will use a qualitative abstraction of such quantitative levels. For defining such abstractions, we have introduced four fuzzy set: *high*, *medium*, *low* and *null*, to which every indicator has a degree of membership. For example, if the indicator of propensity to spend is equal to 0.7, we have:

Membership(0.7, High) = *0.8*
Membership(0.7, Medium) = *0.7*
Membership(0.7, Low) = *0.2*
Membership(0.7, Null) = *0.08*

So, the degree of membership for the fuzzy set "High propensity to spend" is 0.8, and so on.

Fuzzy membership (we will use μ to denote membership in the rest of the paper) will be exploited by the personalization strategies to infer the most appropriate suggestions to be presented to the user.

4.2. The Behavior User Model

As psychological studies show [26], [2], user's actions are the reflection of her motivations and intentions, which derive from the awareness of her needs and wishes. Therefore, observing the sequence of user's actions during the interaction with the system, we can infer her intentions and desires with a reasonable degree of certainty. This means that user actions can be considered as an important feedback about her interests and needs. Moreover, such feedback can be detected by the system and exploited to update previous inferences about user characteristics stored in her UM.

We decided to consider the following types of user actions:

1. *GPS coordinates.* If the user reaches a place, we can determine her GPS coordinates and we can use this information to infer her activities. Unluckily, this feedback suffers from a high degree of uncertainty. In fact, we may consider the following problems:
   - Accurateness. Two places close to each other have similar coordinates and the GPS receiver might not discriminate between them.
   - Start up. Sometimes, when a GPS receiver is switched on, it could take a few minutes to localize the position of the user. If the user turns off and on the GPS receiver in places far from each other, this latency time could increase. Moreover, if the turning on takes place while moving, the GPS could not work properly.
   - Reception. Many GPS receivers have reception problems indoor and/or in particular areas not well covered by GPS signals.
   - Parking. In the case of an on-vehicle GPS receiver, we have to consider the possibility that the user parks her car far from the actual place she is going to.

   The GPS coordinates are a relevant feedback but, considering the previously mentioned problems, it should be associated to other kinds of feedback, in order to improve its reliability.
2. *On-line booking.* From this user action the system can infer, with a high degree of certainty, the user intention to do something.
3. *Route information.* The user request of how to reach a place is a strong signal of his intention to go there. We have to consider that the request for information is not an evidence that the intention will become a real action. So, the relevance of this feedback increases if it is associated with a GPS reception and/or an on-line booking.
4. *Page bookmark* and *page printing.* These are weak feedbacks about user intention, but strong ones about user interest in the page topic.
5. *Item selection.* It is a weak feedback both for user intention and for user interest. In fact, the user can click an item only for curiosity.
6. *Time of permanence on a page.* This is a feedback that can be used in a negative way. In

fact, it is not important how long a user keeps a window open (she could do something else leaving the window open or iconized), but if she immediately leaves a non previously seen page (e.g., pressing the Back button), we can weakly infer a lack of interest in the page topic.

Each feedback communicates user intentions, desires and needs with a different level of strength. We merge these strengths so that each type of action detected by the system has an associated *weight*, that represents the potential impact of the action on the UM. For example, on-line booking is a stronger signal of a user preference than page bookmarking, so the weight associated to the former is higher than the weight associated to the latter.

In order perform dynamic UM update, the first step is to record user actions. In particular, for each user, we store:

- the number of sessions from the last update of her Current UM;
- a time stamp of the current session;
- the action performed by the user, the item selected (if any), the relevance (weight) of the feedback.

When the user has performed a sufficient number of interactions with the system, the inference engine starts working. The number of user's interactions is a crucial issue that must be taken into account. In fact, in order to have a reliable relevance, it is better to activate the update process after a considerable number of user interactions. But, at the same time, if the number of sessions considered before the update is too high, the system might appear as too low and stupid, possibly presenting information of no interest for the user many times, with a high risk of churn. Furthermore, we have to consider users that rarely use the system. In fact, if we start the update process after a lot of interaction sessions, they never see any change. In the current prototype we considered five sessions a good compromise and, in any case, not later then two months from the last update (in the case there is at least one interaction).

Notice that the update process runs off-line, because, as [29] say, it is not a good practice to make any changes during the same navigation session, since it can cause confusion to the user.

Starting from user's feedbacks, a set of rules create a new User Model, the *Behavior User Model* (UM_B) through some steps.

As a first step, a set of rules is activated to determine the features in the User Model on which feedback have an impact. For example, the category of the selected hotel impacts on the user propensity to spend. Thus, the knowledge base contains rules such as:

IF ((HotelCategory = FourStars)
AND (HotelServices = SwimmingPool)
OR (...))
THEN PropensityToSpend = 0.8

where the antecedent refers to properties of the selected item and the consequent contains a couple of values: a feature of the User Model and an indicator. For example, the rule above specifies that whenever the user selects a four stars hotel with swimming pool (or other top class facilities), this gives the system a feedback that the level of her propensity to spend is 0.8.

Notice that UM_B features are a subset of UM_C features, as they include the inferred ones only, i.e., the specific interests and the propensity to spend (see section 4.1).

Moreover, each feedback has an associated relevance (weight), depending on the type of feedback. Thus, from each feedback, we get a list of triples

(feature, indicator, relevance).

For example, we might have a set of triples such as:

(propensity to spend, 0.8, 0.2)
(pop music interest, 0.2, 0.8)
(pop music interest, 0.5, 0.1)

As a second step, the indicators are mapped to the corresponding fuzzy sets (see Section 4.1). For each triple, we compute the membership of the indicator in the four fuzzy sets, obtaining extended triples of the form:

$$\begin{aligned}&(\textit{feature},\\ &\quad\{\ \mu(\text{indicator, High}) = \textit{value}_1\\ &\quad\ \ \mu(\text{indicator, Medium}) = \textit{value}_2\\ &\quad\ \ \mu(\text{indicator, Low}) = \textit{value}_3\\ &\quad\ \ \mu(\text{indicator, Null}) = \textit{value}_4\ \},\\ &\textit{relevance})\end{aligned}$$

where $value_i$ represents the degree of membership to the fuzzy set, and *relevance* represents the weight of feedback. For example, the triple

(pop music interest, 0.2, 0.8)

is transformed into

(pop music interest,
{ μ(0,2, High) = *0.1*
μ(0,2, Medium) = *0.3*
μ(0,2, Low) = *0.9*
μ(0,2, Null) = *0.2* },
0.8)

Finally, we have to merge the triples referring to the same feature; to this purpose, we combine the memberships using the relevance to weight each contribution. For each fuzzy set we apply the following formula:

$$\mu_{UM_B}(\text{Set X}) = \sum \frac{\mu(indicator, X) * relevance}{\sum relevance}$$

Thus, for each feature, we have a result like the following:

{ μ_{UM_B}(pop music interest, High),
μ_{UM_B}(pop music interest, Medium),
μ_{UM_B}(pop music interest, Low),
μ_{UM_B}(pop music interest, Null)})

The set obtained in this way represents the content of the UM_B, inferred by the system from the user's behavior.

4.3. Update of the User Model

The process of updating the user's profile requires a merge between the Current UM and the Behavior UM. The higher is the number of user interactions with the system, the higher is the number of updates performed on the Current UM, and the greater is the "weight" of the Current UM with respect to the Behavior UM. In other words, the "older" is the Current UM, the greater is its impact in the merging, and the smaller is the impact of the Behavior UM. This consideration is crucial in order to achieve the goal of reaching a stable Current UM, that represents the user with a high degree of precision.

Each User Model is composed as follow (one for each feature):

{ μ(feature, High),
μ(feature, Medium),
μ(feature, Low),
μ(feature, Null)})

Thus, we have to combine the fuzzy sets in the Current UM with the corresponding ones in the Behavior UM. To do this, we use the formula below, which updates the Current UM:

$$\mu_{UM_C}(F, Set) = \frac{\mu_{UM_C}(F,Set) * W(UM_C) + \mu_{UM_B}(F,Set) * W(UM_B)}{(W(UM_C) + W(UM_B))}$$

where $\mu_{UM_C}(F, Set)$ represent the membership in the fuzzy set Set for the feature F in the Current UM and $\mu_{UM_B}(F, Set)$ represent the membership in the fuzzy set Set for the feature F in the Behavior UM; *W(UM_C)* is the weight of the Current UM, while *W(UM_B)* is the weight of the Behavior UM. Both of them are numbers between 0 and 1. During the first update, the Behavior UM has a higher weight, than it decreases with the number of interactions.

5. Adaptation Strategies

In this section we describe the strategies implemented in UbiquiTO knowledge bases for performing the different forms of adaptation.

5.1. Adaptation of the Content

The goals of UbiquiTO regarding content adaptation are:

1. Recommending items (places to visit, restaurants, accommodations and so), according to the user model and context (location);
2. Adding recommendations of related places, based again on the user model and location;
3. Personalizing the detail level of items descriptions.

As regards items recommendation, the Recommender computes a score for each item, excluding those that are not compatible with the user time availability (e.g. restaurant closing day, museums opening time, etc.), then it orders the list of items according to the score. The computation of the score takes into account:

- a score depending on the user's interest (provided by the UM) in the category the item belongs to;
- a score depending on the proximity of the item to the user position.

The first score is calculated by a set of rules that match UM dimensions with items features (see knowledge bases in section 3.1). The following is an example of the structure of this type of rules:

IF (logical combination of user features
(Membership to fuzzy sets))
THEN (score for item features)

Consider, for example, the user John, described in section 2: UbiquiTO suggests him to go to an elegant and traditional restaurant, rather than to a fast food, because it makes the assumption that John prefers this kind of places, given his high money availability and gastronomic interests. The applicable rule for John is the following:

IF ((μ(propensityToSpend, High) > 0.7)
AND (μ(InterestInFood, High) > 0.8))
THEN ((TraditionalRestaurant = 0.7)
AND (PlaceMoodElegant = 0.9)
AND (CostHeigth = 0.6))

A score can thus be associated with each item in the database, based on its features and on the scores assigned to features by the rules. A second set of rules gives a score to each item according to the distance from the user location (sub-section 5.1.1 will be devoted to the explanation of how positioning data are detected and exploited).

Finally, the total score is a weighted average of the two previous scores, calculated by the following formula:

$$\frac{Score(UM) * 0.5 + Score(Location)}{2}$$

Thus proximity has a stronger impact on the overall score with respect to user preferences.

Regarding the second goal of content adaptation mentioned above, other proposals, related to the suggested items, are decided on the basis of the User Model and location. The idea is that when the user selects an item, the Recommender provides her a list of suggestions (associated items). For instance, if a young user who likes going out and drinking selects a Mexican restaurant, a list of trendy bars for the 'aperitivo' (before dinner) and wineries (after dinner) close to the restaurant are suggested.

Thus, two kinds of rules are fired. The first one produces a list of places associated to the items, on the basis of: class, category, closing day, opening time, location of all the items. Then, a second set of rules, similar to those explained for items recommendation, selects places that fit user features.

The third goal of content adaptation concerns the personalization of the detail level of items descriptions. A set of rules determine the optimal detail level for the description of each category of items based on the user interests in the corresponding categories. For example

IF ((ItemCategory = Museum)
AND (ItemGenre = AncientArt)
AND (UserInterest μ(Art, Hight) > 0.7))
THEN (DetailedDescription = True)

5.1.1. User location

User location is one of the dimensions used to personalize services to the user. The Context Module manages the localization in three ways:

- *Non-automatic localization.* The user has the possibility to inform the system about her position by selecting a POI (Point Of Interest: an immediately recognizable tourist attraction) on a sensitive map or from a list of items. This method has two main reasons: allowing users which are not equipped with Wi-Fi or GPS receiver to access UbiquiTO services, and to acquire location information independently of the user's current position (e.g., if the user wishes to have some information about a restaurant close to a given place, while she is at home).
- *Wireless LAN.* If the user mobile terminal is equipped with a Wi-Fi receiver and enters wireless modality, her position can be computed on the basis of the signals received from the different access points within the area [24].
- *GPS.* In this case the tourist's device contains a GPS receiver that enables the system to calculate her position with great accuracy. The automatic localization is implemented by a client-server application, that sends the GPS coordinates to the web server. At the server side, the Watcher (actually, its sub-module called ubiStorer, see Section 3) stores them and makes them available to the Context Module. This module cooperates with the Recommender to suggest the results on the basis of their distance (see the previous Section). At the client side we have Ubi-

Tracer, the module which communicates with the GPS device (by the serial port) and sends the coordinates to the server, by TCP/IP network socket, every time slice. In order to allow the automatic localization, the UbiTracer client must be running in background during the navigation. For details regarding platform and environment for mobile devices, see the Implementation Remarks at Section 8.

Independently of the method used, the user position is represented by a pair of coordinates. Given this piece of information, the Context Module retrieves, from the Places DB, the coordinates of places to be recommended and calculates the distance between user position and every place. On the basis of these results, the Recommender agent assigns a score to each item (see the description above).

5.2. Adaptation of the User Interface

As described in the previous section, various aspects of the user interface must be adapted, not only to the user needs, but also to the usage context, and in particular to the specific device used to access the services and information offered.

As suggested in the literature, adaptation techniques can be exploited to effectively handle the interaction in mobile user interfaces. For instance, [5] and [38] suggest that adaptation techniques can be successfully applied to enhance the usability of mobile and wireless services. Therefore, we considered their suggestions in order to improve the usability of our mobile interface.

In the current prototype of UbiquiTO, two User Interfaces (UI) have been designed: a desktop and a PDA/smartphone UI. The initial Information Architecture [35] process has been similar for both UIs: we identified a set of categories and subcategories grouping the items (tourist attractions, restaurants, accommodations, and so on) and corresponding to services and contents offered by UbiquiTO. The categories have been mapped onto the menu voices of global navigation and the subcategories onto the menu voices of local navigation. For the desktop UI, we adopted a tab-menu approach, with global navigation in the top horizontal area of the page, the local navigation in the lefthand side of the page, and the contextual items list in the central area (see Figure 2). Concerning the PDA/smartphone interface, we simplified the interaction (see Figure 3):

1. the main categories are shown in the home page (global navigation);
2. when the user selects a category, the corresponding subcategories list is presented in the next page (local navigation);
3. when the user selects a subcategory, the item list is shown in the next page (contextual navigation).

Since the final item list (point 3) is ranked according to the user profile and location, we added, as adaptive annotation [6], a symbol to express the concept of system personalization. In particular, we wanted to communicate to the user *i)* the personalization of the presentation of the items, and *ii)* the strength of recommendations. To decide which type of symbols best represent these concepts, we performed an experimental evaluation with 34 real users (for details see [10]). The quantitative and qualitative results we gained highlight the fact that a good solution to communicate the system recommendations is represented by the emoticons. Therefore, the system associates to each recommended item an emoticon representing a smiling face, a neutral face or a sad face, depending on the rate associated. Given the item list, the user can select a specific item, to get more information about it. In the PDA/smartphone UI, the quantity of information presented to the user is reduced with respect to the one shown in the desktop UI: every item description offers only the most relevant information in a compact format.

The agent in charge of the adaptation of the presentation (Presentation Adapter) performs the following actions, by exploiting a set of adaptation rules:

1. it automatically detects the user's device;
2. it selects the most appropriate layout on the basis of the device characteristics;
3. it selects the amount of information to be displayed, according to both the screen size of the device and the user interests;
4. it changes font size and background color according to context conditions (e.g., time of the day, movement or not) and user special needs.

The detection of the type of device exploited by the user is performed by the Watcher, that receives data encoded within the HTTP request from the client. On the basis of such data, the Device Module identifies the device characteristics (e.g.,

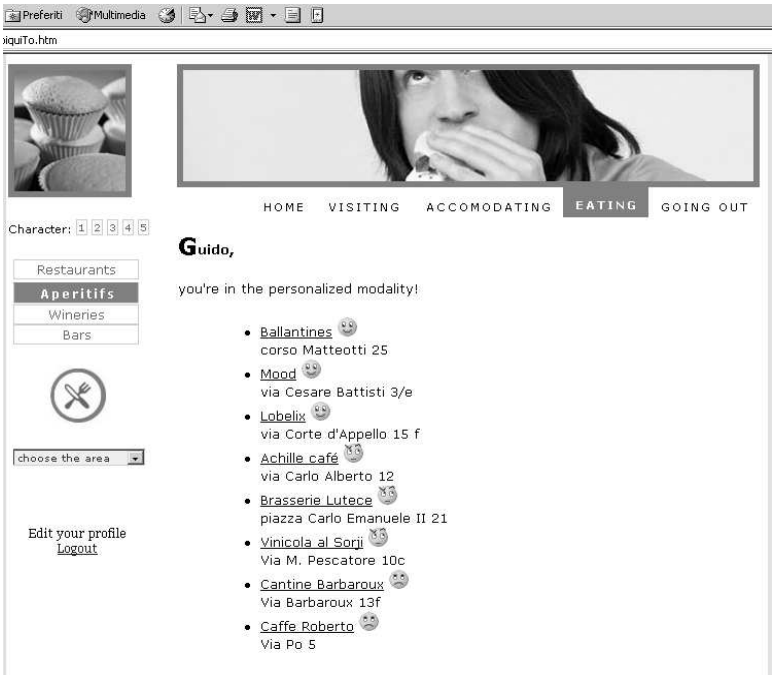

Fig. 2. The desktop interface.

type of device, operating system and web browser) needed to adapt the various aspects of the user interface.

The dynamic generation of the user interface is performed by the UI Generator module, that apply an XSL stylesheet to the XML object that represents the personalized content.

The rules that select the most appropriate layout, contain conditions on the type of device and select a specific XSL stylesheet as a consequence.

The other two sets of rules (those for selecting the amount of information and those defining fonts and colors) are encoded within the stylesheets. In particular, the rules that select the amount of information to be displayed evaluate both the screen size and the user interests and select one of the four stored versions of the item description: long/essential, long/detailed, short/essential and short/detailed (see Section 3.1).

The rules that define font size and background color check context conditions, like, for instance, the time of the day (which is an indicator of the possible light conditions), together with some specific user characteristics, like her age (which implies possible vision impairments); on the basis of the evaluation of these conditions different HTML tags and attribute values are included in the we page.

6. Testing

In the User Modeling community the importance of systems evaluation has been strongly advocated [12] and now it is a shared principle. As underlined by Brusilovsky et al. [7], a successful evaluation of an adaptive system should be decomposed into different layers, corresponding to the main adaptation constituents. Therefore, we performed a layered evaluation by testing separately the effectiveness of UI adaptation (for details see [10]) and content adaptation. In order to test this aspect, we evaluated the system recommendations, based on the system inferences about users preferences. Thus, we directly asked a group of users about their real preferences and, then, we compared them with the corresponding system recommendations.

To obtain reliable users data we distributed a questionnaire to 70 users (40 males and 30 females, 26-45 aged), identified following a proportional layered sampling strategy[1]. Three main topics were identified in the questionnaire:

[1]In the proportional layered sampling strategy the population is divided into layers, related to the variables that have to be estimated, and containing each one a number of individuals proportional to its distribution in the target population.

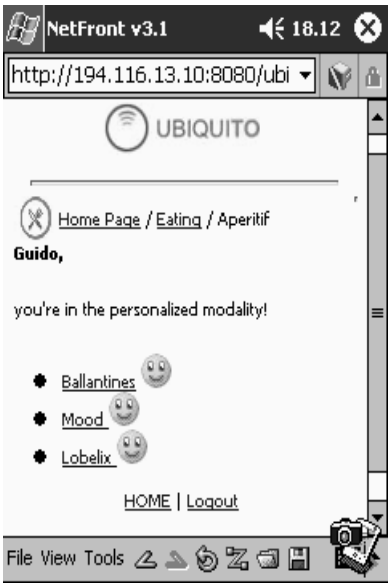

Fig. 3. The PDA interface.

1. personal data (sex, age, job);
2. general interests (music, art, movies, food);
3. specific interest (classical music, pop music, rock music, antique art, modern art, contemporary art, history, natural science, comedy, dramatic movie, fantasy, traditional food, ethnic food, fast food, pizzerie).

The final questionnaire included 21 questions. To avoid any possible interviewer's interferences, the questionnaires were sent by email and filled in autonomously by the users. The questionnaire was anonymous and introduced by a written presentation explaining the general research aims. For the questions concerning personal data, the participants were required to thick the appropriate answer from a set of given alternatives. In the other questions, users had to express their level of agreement with the options by choosing an item of a 3-point Likert scale.

We provided the system with the personal data and general interests (points 1 and 2) to generate both the inferences for the UM initialization and final recommendations. Then we calculated the distance between the real user preferences collected by the questionnaire (specific interests, point 3) and the corresponding system recommendations. We exploited a statistical accuracy metric, the MAE (Mean Absolute Error) [37], which evaluates the distance between the system predictions and the collected users opinions by means of rate vectors. Obviously, higher values mean worst recommendations.

The average MAE results were 0.164, with 20 users with a MAE between 0.038 and 0.168, and 10 users with a MAE between 0.184 and 0.292. We considered the average results satisfactory[2], since this evaluation does not take into account the dynamic update of the User Model (which will be evaluated in a second step) and a MAE equal to 0.164 can be considered a good result for a first user-system interaction.

7. Related work

Mobile guides typically merge technologies and approaches from different fields. UbiquiTO, in particular, is an adaptive system borrowing techniques from the User Modeling research area and combining them with wireless technologies, in an outdoor tourist application domain. Consequently, several works in different areas are significantly related to the project.

Within our approach, User Modeling is exploited to manage user profiles, which are the basis for tailoring content, presentation and interface

[2]Good et al. [19] suggest that, in the evaluation of a recommender system, a satisfactory value of MAE should be about 0.7, in a range of 1-5, therefore 0.14 in a range of 1-3.

to the user's characteristics. As far as the adaptation of the content is concerned, the wide research area on recommender systems is relevant. A survey of the entire field is out of the scope of the present paper: some good surveys can be found in [33], [17], [34]. Tailoring the presentation of the information to user's interests and needs is one of the main issues within the Adaptive Hypermedia research area. [6], [20], [25] are reference surveys of the filed.

A second type of adaptation implemented by the UbiquiTO system is User Interface adaptation, that includes the automatic generation of the different UIs depending on user interests/needs and on the device used to access the system. The issue of automatically generating different UIs in order to adapt to different devices has been faced within many projects (e.g., [21], [16], [30], [13], [28]). All these projects share with UbiquiTO the idea of having a (abstract) definition of a UIs that is exploited by specific modules to generate user interfaces that take device features into account. For instance, within the Princess Project [21] the provision of multimedia services on different devices is achieved by generating different UIs (e.g. HTML, WML, SM, e-mail) that take into account several characteristics of the device, such as I/O modality, supported languages, quality of service, communication protocols, security, mobility. [16] presents an approach based on three main concept: abstract definition of the UI; priority values, assigned to UI elements; multi-level stylesheets to generate the final UI. [30] developed an information system for the Motorola museum in Illinois that exploits XSL Transformations to generate WML and HTML from and XML object. Dygimes [13] and TERESA [28] are based on the definition of a task model of the application that represents a common source to generate specific UI.

Finally, UbiquiTO adapts the interaction to the user's location, which is traditionally considered by context-aware applications, which range from the first experiments (e.g. the Active Badge [40]) to the more recent mobile tourist guides, mentioned in the following of this section.

With respect to the research carried on within the mentioned areas, we think that the most important characteristic of UbiquiTO is its flexibility, for example, the possibility of the system to adapt to several factors, integrating the adaptation to device and location with adaptation to the user characteristics. This combination enables the tourist to use her own mobile terminal, not necessarily equipped with specific positioning devices or client-side applications, and to benefit from the advantages of information filtering, based on her interests, preferences and context conditions, implicitly and unobtrusively monitored and updated.

In this perspective, the projects that are more directly related to our approach are tourist mobile guides, such as Cyberguide [1], Guide [11], Lol@ [32], Crumpet [31], Real [4], SmartKom [39], Deep Map [23], TellMaris [15], to cite only some examples.

In particular, the most relevant related project is Crumpet [31], a tourist guide for the cities of Heidelberg, London, Helsinki and Aveiro. The main features are the personalization of services to the user's current location, personal interests, and history of interaction with the system; the possibility to access the Crumpet services using PDA and smartphones; the adaptation of content presentation to changing technical environments; localization using GPS or other operator based technologies (e.g. GSM, UMTS). It offers a very wide range of services, such as a personal tourist guide, the visualization of the shortest way to a point, proactive suggestions, etc. A final relevant aspect is the agent-based architecture, conforming to FIPA standards.

Guide [11] is another relevant related project. It is a tourist guide for the city of Lancaster in Great Britain. It is adaptive with respect to the location of the user, her walking speed, the places already visited, the time of the day and the language and interests of the user. This last information is acquired using a registration form and is not updated or refined during the interaction. It uses network cells and interaction with the user for positioning and offers a great deal of navigational and guiding services. The main lack of the system concerns the client device: Guide services may be accessed only using an ad hoc terminal rented at the Lancaster tourist office. On the contrary, Lol@ [32], a guide for the city of Vienna, is adaptive toward the device, but not toward the user. It uses GPS as positioning technology (beside network solutions in the next version) and makes use of a GIS system for the generation of the map. As in UbiquiTO, the layout is adapted using XSL stylesheets. Despite of the complex infrastructure of this system, as emphasized by [5], the lack of adaptivity (and

especially implicit adaptivity) toward the user interest, preferences and goals can compromise the possibility to enjoy the services in mobile environments. However, apart for Crumpet, the majority of the systems which are adaptive toward the user features use a static User Model (see for example Deep Map [23], Real [4], and the above described Guide).

In term of flexibility, one last feature we would like to briefly discuss is the modality to elicit the user position. Roughly one half of the outdoor systems use GPS, while just a few allow the estimation of the position by interacting with the user. Guide, Lol@, Real and Deep Map are some examples, even if they differ in the way to achieve that goal [22]. Real and Guide simply use the selection on a static list/map, Lol@ allows the user to make a selection on a dynamic list based on the last known position of the user. Finally, Deep Map includes a sophisticated model, based on position history, situational knowledge etc.

With respect to this aspect, one of the goals of UbiquiTO is to enable a multi-modal positioning. In the current prototype it offers three modalities: layered maps of selection, WiFi, and GPS modality. A relevant aspect of the layered map is that it is built upon the concept of points of proximity which shares the principles of landmarks, already experimented in several contexts in order to provide information. The main idea is that they represent relevant points in the user mental map and help the user to construct a mental representations of unfamiliar environments (see [27] for more details).

8. Implementation remark

We implemented our system by using the Java Servlet technology, in a Tomcat environment. The inference engine is based on Jess (Java Expert System Shell, a Java library) and Fuzzy Jess (a Java API that provides a fuzzy reasoning capabilities to Jess). Users and places data are stored into MySQL databases. We use Superwaba (a virtual machine for PDA) in order to trace users that connects to the system with a PDA or with a smartphone. Both the SuperWaba environment and the UbiTracer client can be automatically downloaded and installed by the browser in the user device.

Regarding the dynamical generation of different user interfaces, the system produces an XML document which represents the content of the interaction. Then, it exploits XSLT to transform the an XML document into other formats (HTML, XHTML, etc.), depending on the device.

9. Conclusions

In this paper we presented UbiquiTO, an expert tourist guide for mobile users that adapts the content provided and the interaction to the user interests and to the device used, as well as to the physical location and other context conditions. UbiquiTO combines User Modeling techniques with wireless technologies, thus the integration of different adaptation strategies is the most relevant aspect of the project.

Many aspects will be improved in future versions. Some improvements has the objective of further enhancing system flexibility. For instance, we would like to exploit multi-modal interaction and to take into account user's goals.

Moreover, besides PC, PDA, and smartphone, other devices will be taken into account (e.g., onboard equipments, digital TV sets), a larger set of services will be included, and we would export UbiquiTO framework to other Italian cities.

Finally, as far as the evaluation of the system is concerned, we are planning to perform a second evaluation, which includes the dynamic update of the UM. Moreover, we will test the system with real usage data, when UbiquiTO services will be linked to the local government web site.

Acknowledgements

Special thanks to Andrea Toso for the support and management of the project. We are very grateful to Ilaria Amendola, Andrea Crevola, Valeria Grosso, and Monica Perrero for having helped us during the development of the project and for their fruitful suggestions. We also thank all the CSP staff for their participation in the tests.

References

[1] Abowd, D. A., Atkeson, C. G., Hong, J., Long, S., Pinkerton, M.: Cyberguide: A Mobile Context-Aware Tour Guide. Wireless Networks 3(5) (1996) 421-433

[2] Allport, G. W.: Pattern and growth in personality. Rinehart and Winston, Holt, NY (1965)

[3] Amendola, I., Cena, F., Console, L., Crevola, A., Gena, C., Goy, A., Modeo, S., Perrero, M., Torre, I., Toso, A.: UbiquiTO: a Multi-Device Adaptive Guide. Proceedings of conference "MobileHCI'04", Sept. 13-16, Glasgow, Scotland (2004) 409-114

[4] Baus, J., Kray, C., Kruger, A.: Visualization of route descriptions in a resource-adaptive navigation aid. Cognitive Processing 2(2-3) (2001) 323-345

[5] Billsus, D., Brunk, C., Evans, C., Gladish., B., Pazzani, M.: Adaptive Interfaces for Ubiquitous Web Access. Comm. of the ACM 45(5) (2003) 34-38

[6] Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 6(2-3) (1996) 87-129

[7] Brusilovsky, P., Karagiannidis, C., Sampson, D.: The benefits of layered evaluation of adaptive applications and services, S. Weibelzahl, D. N. Chin, and G. Weber (Eds.), Empirical Evaluation of Adaptive Systems. Proceedings of workshop at UM2001 (2001) 1-8

[8] Brusilowsky, P.: Adaptive Hypermedia, User Modelling and User Adapted Interaction 11 (1-2)(2001) 87-110

[9] Camera di Commercio Industria Artigianato e Agricoltura di Torino: Turismo a Torino: perche' si soggiorna e quanto si spende. http://images.to.camcom.it/f/Studi/Sp/Spesa_Extra_alberghiera_To.pdf, Turin (2003)

[10] Cena, F., Gena, C., Modeo, S.: An adaptive tourist guide in mobile context and its evaluation. Technical report, February 2005, http://www.di.unito.it/ cgena/pub.html.

[11] Cheverest, K., Davies, N., Mitchel,l K., Smyth, P.: Providing tailored context-aware information to city visitors. In Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2000). LNCS Vol. 1892, Springer-Verlag, Berlin Heidelberg New York (2000) 73-85

[12] Chin, D.N.: Empirical evaluation of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, 11(1-2)(2001) 181-194

[13] Conino, K., Luyten, K., Vandervelpen, C., Van den Bergh, J., Creemers, B.: Dygimes: Dynamically Generating Interfaces for Mobile Computing Devices and Embedded System. LNCS Vol. 2795, Proc. of Mobile HCI 2003 Springer-Verlag (2003)

[14] Console, L., Torre, I., Lombardi, I., Gioria, S., Surano, V.: Personalized and Adaptive services on board a car - An application for tourist information: *J*ournal of Intelligent Information Systems, Vol 21, No 2 (2003) 249-284

[15] Coors, V., Gjesdal, O., Sulebak, J. R., Laakso, K.: 3D Maps for Boat Tourists. In Frew A., Hitz M., O'Connor P. (eds.). Information and Communication Technologies in Tourism. Proc. of ENTER 2003, Springer Computer Science, Wien New York (2003) 298-305

[16] Dees W.: Device Independent User Interfaces. Proc. of the 12th Int. WWW Conf. Budapest, Hungary (2003) 209-249

[17] Fink, J., Kobsa, A.: A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. User Modeling and User-Adapted Interaction, Special Issue on Deployed User Modeling, 10(2-3) (2000) 209-249

[18] Gena, C., Torre, I.: The Importance of Adaptivity to Provide On-Board Services. A Preliminary Evaluation of an Adaptive Tourist Information Service on Board Vehicles Special Issue on Mobile A.I. in Applied Artificial Intelligence Journal (2004)

[19] Good, N., Schafer, J. B., Konstan, J. A., Botchers, A., Sarwar, B. M., Herlocker, J. L., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations, Proc. of the Sixteenth National Conference on Artificial Intelligence (1999) 439-446.

[20] Kobsa, A.,Koenemann, J., Pohl, W.: Personalized Hypermedia presentation techniques for improving online customer relationships. The Knowledge Engineering Review, 16(2) (2001) 111-155

[21] Korva, J., Plomp, J., Maatta, P., Metso, M.: Personalized Hypermedia presentation techniques for improving online customer relationships. Proc. of the 2nd Int. Conf. on Mobile Data Management, LNCS Vol. 1987, Springer-Verlag, London, UK (2001) 252-259

[22] Kray, C., Baus, J.: A survey of mobile guides. Workshop on HCI in mobile guides, 5th Int. Symposium on HCI with Mobile Devices and Services. Udine, Italy (2003)

[23] Malaka, R., Zipf, A.: DEEP MAP - Challenging IT research in the framework of a tourist information system. Information and Communication Technologies in Tourism. Proc. of ENTER 2000, Springer Computer Science, Wien New York (2000) 15-27

[24] Marchioro, C.E., Ghittino, A., Ferri, D., Borri, R., Dovis, F.: Wireless Campus project. TERENA Networking Conference 2003. Zagreb, Croatia (2003)

[25] Maybury, M., Brusilovsky, P.: The adaptive Web. Communications of the ACM, 45(5) (2002)

[26] Maslow, A.H.: Motivation and Personality, Harper and Row, New York, (1954)

[27] Michon, P. E., Denis, M.: When and why are visual landmarks used in giving directions. In: Montello, D.R. (ed.): Spatial Information Theory. Lecture Notes in Computer Science, Vol. 2205, Springer-Verlag, Berlin Heidelberg New York (2001) 292-305

[28] Mori G., Patern F., Santoro C.: Tool Support for Designing Nomadic Applications. Proc. of the 8th Iin. Conf. on Intelligent User Interfaces, ACM Press, New York, NY, USA (2003) 141-148

[29] Nielsen, J.: Designing Web Usability: The Practice of Simplicity, New Riders Press (1999)

[30] Pashtan, A., Kollipara, S., Pearce, M.: Adapting Content for Wireless Web Services. IEEE Internet Computing (2003)

[31] Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P., Zipf, A.: CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism. 3G Mobile Communication Technologies. London, UK (2001) 26-29

[32] Pospischil, G., Umlauft, M., Michlmayr, E.: Designing LoL@, a Mobile Tourist Guide for UMTS. Information Technology and Tourism 5(3), Cognizant Communication Corporation (2003) 151-164

[33] Resnick, P., Varian, H.R. (eds): Special Issue on Recommender Systems. Communications of the ACM, 40(3) (1997)

[34] Riecken, D. (ed): Special Issue on Personalization. Communications of the ACM, 43(8) (2000)

[35] Rosenfeld, L., Morville, P.: Information Architecture for the World Wide Web. O'Reilly (1998)

[36] Salton, G., McGill M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1984)

[37] Shardanand, U., Maes, P.: Social Information Filtering for Automating "Word of Mouth". Proceedings of CHI-95, Denver, CO. Strauss, A. L. and Corbin, J. M.: Basics of qualitative research: techniques and procedures for developing grounded theory, SAGE, Thousand Oaks (1998)

[38] Smyth, B., Cotter, P.: The Plight of the Navigator: Solving the Navigation Problem for Wireless Portals. Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Systems. Malaga, Spain (2002) 328-337

[39] Wahlster, W., Reithinger, N., Blocher, A.: SmartKom: Multimodal Communication with a Life-Like Character. 7th European Conference of Speech Communication and Technology (Eurospeech 2001), Vol. 3, Aalborg, Denmark (2001) 1547-1550

[40] Want, R., Hopper, A., Falcao, V., Gibbons, J.: The Active Badge Location System. ACM Transaction on Information Systems, 10(1) (1992) 91-102

[41] Weibelzahl, S., Weber, G.: Evaluating the Inference Mechanism of Adaptive Learning Systems. In Brusilovsky, P., Corbett, A. and de Rosis, F. (Eds.) User Modeling: Proceedings of the Ninth International Conference (pp. 154-168) (Lecture Notes in Computer Science LNAI 2702; (c) Springer-Verlag). Berlin: Springer (2003)