Supporting Personalized Interaction in Public Spaces

Giovanni Cozzolongo, Berardina De Carolis, Sebastiano Pizzutilo
Intelligent Interfaces
Department of Informatics
University of Bari
{cozzolongo, decarolis, pizzutilo}@di.uniba.it

Abstract

In this paper, we describe an architecture supporting personalized interaction with information services available in public spaces. In particular, we focus the presentation on the personal user modeling agent that runs on the user mobile device and is able to exchange data with the environment in order to get personalized information.

Keywords

User modeling agents, embodied conversational agent.

INTRODUCTION

Interaction with information services available in public spaces may be provided in different ways. Usually, it can be accessed using public kiosks, LCD displays or, if the user has his/her personal device, this can be used to interact with the environment in a *private* way. However, this does not mean that the interaction is personalized in order to meet the requirements and interests of the user. Then, the physical space has to be designed so as to 'sense' the particular users being in the environment and to 'know' their interests and preferences, the environment can then use this information to create more specifically targeted presentations [4, 8].

With this aim, there are two possible approaches to user modeling: in the first one, this is one of the tasks of the personalization component of the smart environment, in the second one, the user has his/her user model on the personal device. In this case, from the architectural point of view, it may be implemented as a "user-trusted" server and may be replicated at the beginning of each interaction with an environment, or, it may stay with the user all the time, either on his or her personal device or on a gadget that the user always wears [14,9].

If this approach is considered, interaction may become personal even in a public space. Users may move around with their own personal device: when in proximity of an information access point a service discovery protocol can be activated, user-related information may be transferred to the environment and, then, requested information may be adapted to that user, who can handle interaction through her/his device or through a public kiosk that is connected to the personal device for user modeling purposes [3].

In this paper, we present a distributed architecture supporting personalization with information services available in public environment. In particular, we investigate how a personal User Modeling Agent (UMA) has been designed and implemented for achieving personalized interaction in case the user interacts with the environment using his/her device.

This agent has to model the user behaviour and to transfer to the Environment Agent what can be inferred about the user in that domain according to required service. Obviously, these two entities have to *understand* each other; at this aim the UMA has to transfer information in a xml-based language compliant to some ontologies. To this aim our Agent uses a simplification of *UbisWorld* language and ontologies [20].

As far as interaction is concerned, it may happen in two ways: i) in presence of environment information display facilities (kiosk, lcd display, etc.), the user may decide to interact directly with environment devices, or ii) using his/her personal device.

In the first case, we use an Embodied Conversational Agent (ECA) as interaction metaphor for representing environment information service [2] and, in this case, the personal device is used in a transparent way in order to provide user modeling services to the environment. ECAs are software agents with a more or less realistic and 'human-like' embodiment, which exhibit many of the properties of humans in face-to-face conversation, including the ability to provide a natural interaction and enforce social involvement [1, 10]. They provide a type of multimodal interface where the agent can dialogue with the user by employing verbal and non-verbal expressions showing human-like qualities such as personality and emotions [1, 16].

In the second case, the information is presented and organized on the user personal device according to a predefined interface structure whose content is generated dynamically.

In a first version of the system we adopted a mixed approach where the information was partially shown on the user device and partially in the public kiosk leaving however the interaction control on the PDA [3]. After an informal evaluation we discovered that this approach was distracting and sometimes confusing the user since he/she had to follow the ECA presentation and then look on the personal device in order to decide what to do in the following interaction move. However, a further evaluation study has been planned in order to test and compare the two approaches.

The paper is structured as follows: after a brief description of the system architecture, we will explain how the Environment Agent and the UMA are designed and implemented by showing an example of its application in a tourist information point. This is just an example domain used to test our approach; the presented architecture has been designed in a domain independent way and can be used to support the interaction with different types of environments. Obviously, domain-dependent knowledge bases have to be implemented according to the environment scope. Conclusions are outlined in the last Section.

SYSTEM ARCHITECTURE

The functional architecture of the system is illustrated in Figure 1. It includes two main components: the Environment Agent, that handles access to service information and public interaction, and the Personal Device Agent running on the user computing device, which handles user modeling and personal interaction. Communication between the two Agents are performed using communication protocols exchanging ACL compliant messages whose content is expressed according to XML-based languages [13].

Let's see in more details how these two components work.

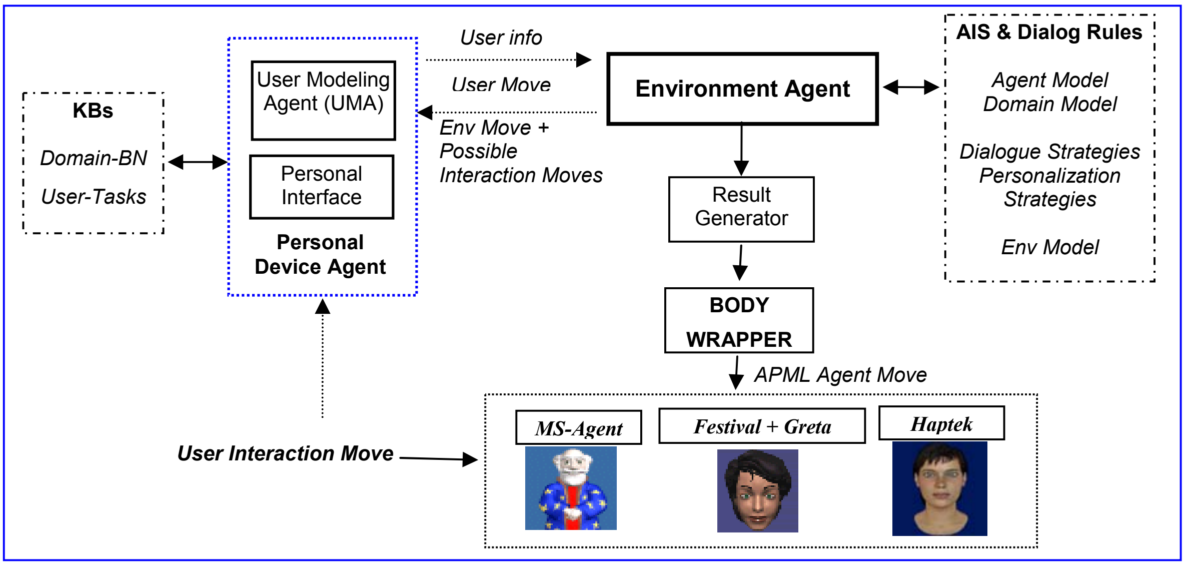


Figure1: Functional Architecture of the Distributed System

The Environment Agent

The Architecture of the Environment Agent is based on the model of a Natural Language Generation (NLG) system [18]. Given a set of goals to be achieved in the selected domain (tourist information service in this case), the Agent plans *what* to communicate to the user and decide *how* to render it according to the expressive capabilities of its "body" and to the conversational context. In this view, the Agent is seen as an entity including two main components (a 'Mind' and a 'Body') which are interfaced by a common I/O language, so as to overcome integration problems and to allow their independence and modularity [6].

Then, in a given phase of the interaction, given a goal, the environment agent selects a suitable plan in a library of plan recipes relative to that domain. The selection is based on the current Agent Internal State (AIS) that stores the following information: i) the environment model (information about the application domain, type of interaction, objects or points of interest in the digital or physical world to which the Agent may refer), ii) the agent model (goals and relations among them, personalization strategies), iii) the user-related information (acquired by the User Modeling Agent during the interaction) and iv) the interaction history.

Let's make the hypothesis that the system is running in a Southern Italy Tourist Information Point providing information about the region. Initially, according to the location and the environment settings, the initial environment and agent models are triggered (Table1).

The Environment model stores information about the type of interaction space, the list of objects and places that can be referenced within the digital space or the real one. This knowledge about the environment enables the agent to use, for instance, deixis to refer to objects mentioned in the interaction.

If the interaction happens through an ECA displayed in a public kiosk then the Agent's physical and behavioural features are set to meet the requirements of the expected target users (adults with interest in visiting the region, knowing about suggested itineraries, local attractions, food and so on) and of the cultural behaviour which is typical of the agency location [7].

These features influence the agent embodiment and behaviour. We selected a dark-haired female face showing a “warm” friendly behaviour. The role of a travel agent does not require a strong empathic attitude; she has rather to establish a social relation with the user by showing that it understands the user feedback.

Table1: an example of AIS in the Tourist Information domain

Environment Features	
Type	Touristic Information Point
Interaction Space	Public
Space References	(infopanel,coord1), (front desk,coord2), …
Agent Features	
Role Personality	Travel Agent Friendly
Gender Culture	Female Southern Italy
Emphatic Attitude	Medium
Default Agent Goals	

Describe(Role(Agent))
Present(touristic-facilities)
∀x | x = Touristic-place:
Suggest(x),DescribeinGeneral(x),
Describe(SelectedItineraries(x)),Describe(Art&Culture(x)),
Describe(Nature(x)), Describe(Accomodation(x))	

The agent’s domain-knowledge (touristic info in this case) is directly related to the environment in which interaction occurs. As a consequence, corresponding default goals and communicative plans of the agent are activated as shown in the last section of Table1: first of all, the agent introduces herself; then, presents tourist facilities etc. During interaction, the agent’s goal can be updated according to explicit users requests.

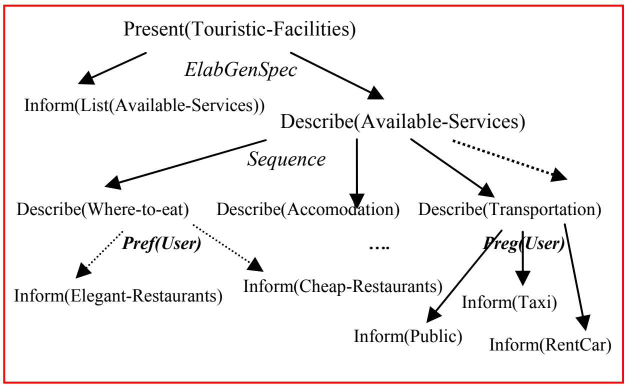


Figure 2: Example of Discourse Plan

Figure 2, illustrate the structure of a plan for *Presenting Tourist Facilities*, which is stored in DPML (Discourse Plan Markup Language) [5]. It first presents general information about the available services, then it describes interactively in details each service. The presentation can be adapted to user preferences by showing the branch of the tree whose focus matches the user’s preference for that category of service. In case user preferences are not accessible or available, the system will present all the possible alternatives.

When the presentation plan that satisfies the current constraints is selected, information to be presented is organized accordingly. The produced result is specified using an extension of APML (Affective Presentation Markup Language: [5]). This resulting string combines the text with tags which specify the ‘meanings’ that the Body will have to attach to each part of it. These meanings include the communicative functions that are typically used in human-human dialogs (emphasis, affective, meta-cognitive, performative, deictic, adjectival and belief relation functions) [15] and the background relative to the information being conveyed.

An example of APML for the *Inform(PublicTransportation(touristic-place))* leave of the plan in Figure2 is shown in the following table:

Table2: an example of extended APML string

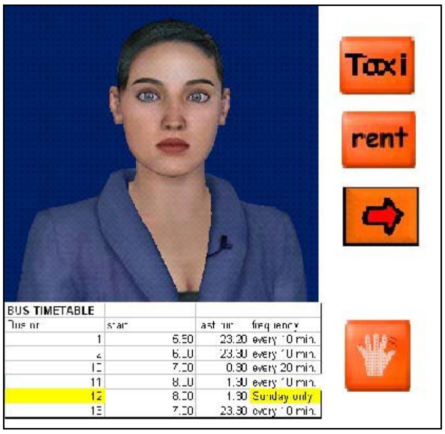
```
<xapml>
<background>
<choice type=”ask-for>
<item>Taxi Information</item>
<item>Rent Car</item>
<item>other</item>
<item>bye</item> </choice>
<focus-info><item>bus-timetable</item>
</focus-info>
</background>
<main>
<performative type="inform">Public
Transportation in Bari: buses runs
<adjectival type=”more-or-less”>
approximately</adjectival> every 30
minutes.</performative>.
<performative type="inform" affect=”sorry-
for”> They are always late </performative>
<performative type=”inform”> More Information
can be obtained by calling to 800600800!
</performative>
</main>
</xapml>"
```

Once the APML string has been generated, the expressed meanings can be rendered according to the interaction modality.

In case the user is interacting with the ECA, the string will be interpreted by the Body Wrapper module in the following way (Figure 3):

- the `<background>` tag specifies the agent’s background information, this can be rendered as a set of buttons, a web page, a table, and so on, according to the specified type. In the previous example, the `<choice>` tag of the type “ask-for” will be rendered as a set of mutually exclusive clickable buttons and a table showing the bus timetable.
- the `<main>` tag specifies the main message, comment or explanation about the background information that will be delivered by the agent. In this phase the Body Wrapper decides which combination of body signals

(verbal and non verbal behaviours) to use to convey every meaning specified in the APML move. In this way, the agent move, tagged at the meaning level, can be coupled with different bodies using a meaning-signal table [6]. So far, we developed APML-wrappers for Greta [15], Haptek [12] and MS-Agent technology.



BUS TIMETABLE			
Bus nr	start	last run	frequency
1	5.50	23.20	every 10 min.
2	6.00	23.30	every 10 min.
10	7.30	0.30	every 20 min.
11	8.00	1.30	every 10 min.
12	8.30	1.30	Sunday only
13	7.30	23.30	every 10 min.

Figure 3: an example of interaction generated from the APML fragment in Table 2.

Table3: another example of extended APML string

```
<xapml>
<background>
<choice type=”ask-for”>
<item>Special Offers</item>
<item>Itinerary</item>
<item>Restaurants</item>
<item>Transportation</item>
<item>bye</item> </choice>
</background>
<main>
<performative    type="greet">Hi!I’m    Maria!
</performative>.
<performative   type="inform"   affect=”happy-
for”>  I’m  here  to  provide  you  tourist
information    about    this    region    of
Puglia!</performative>
</main>
</xapml>"
```

If the user is interacting through his/her device, the APML move will be rendered, according to the structure of the personal interface(Figure 4) in the following way:

- tags specified within the `<background>` that can be used to ask for information will be rendered as a set of clickable buttons (i.e. `<choice>` tag of the type “ask”).
- the `<main>` tag specifies the main message that will be shown in the hyper/textual window of the personal interface together with focus information eventually specified in the background.

This transformation is performed by the Personal Device Agent by applying XSL rules to the APML string.

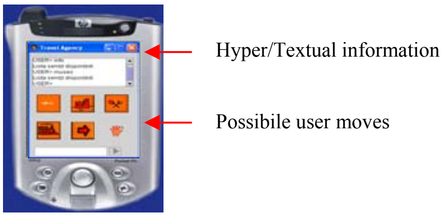


Figure 4: an example of interface on the personal device generated from the APML string in Table 3.

The Personal Device Agent

This component enables the user to personalize the interaction with the environment agent. It includes two sub-components, aimed at modeling the user and, as shown previously, at providing an interface for exchanging messages (user and agent moves) with the environment.

As far as user modeling is concerned, possible approaches to these problems are represented by a centralized, a distributed or a mobile approach [14]. All these approaches present advantages and disadvantages. In traditional client-server information systems, the most frequent design choice is to store the User Model on the server side, by enabling the user to access her/his model after having been recognized by the system. In the distributed solution, user information are stored in different servers, reducing in this way the computational load but presenting problems of redundancy/incompleteness of user information and consistency. In the mobile approach the user "brings" always with her/himself the user model, for instance on an handheld device, and, when the interaction with an environment starts, her/his profile is passed to the environment user modeling component; it seems to be very promising since it presents several advantages. The information about the user are always available and updated, and can be accessed in a wireless and quite transparent way, avoiding problems related to consistency of the model, since there is always one single profile per user.

The architecture of User Modeling Agent (UMA) is based on mobile approach in the following sense:

A Personal Device is used mainly in situations of user mobility. Normally, when the user is in more “stable” environments (i.e. home, office, etc.) he/she will use other devices belonging to that environment (i.e. PC, house appliances, etc.). In this view, the Personal Device can be seen as a satellite of other “nucleus” devices that the user uses habitually in his/her daily life. Then, the UMA has to be able to handle this nucleus-satellite relation.

With this aim, instead of implementing a truly mobile agent, the UMA is cloned and lives on all the user platforms/devices. However, although the chosen approach simplifies the implementation, it requires transferring knowledge needed for user modelling and opens consistency problems in maintaining a global image of user preferences, interests, habitual behaviour, and so on.

In our approach, user models are organized in a hierarchy [19] whose nodes represent relevant interaction environments, task families, interest groups. Each entitiy in the hierarchy represent a subset of user model data relevant to the specified domain, task, etc.

As far as the modeling strategy is concerned, the UMA employs a hierarchical approach to **Belief Network (HBN)** [11] organized as in Figure 5.

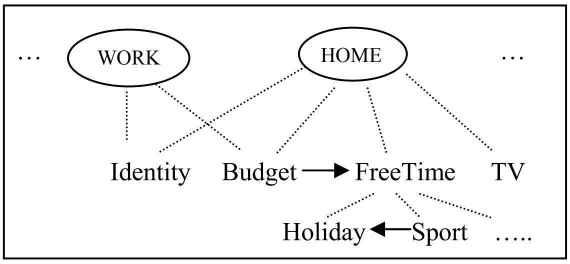


Figure 5: an example of User Model HBN

The roots of the hierarchy represents *user modelling scopes* (interaction environments). Nodes in lower levels of the HBN models specific subset of user data. Dotted lines represent hierachical dependencies while arrows represent a causal link according to [11]. Each node represent a Belief Network (BN) aiming at modelling a default behaviour concerning that domain or task. Each network has a representation of the *context* as nodes that receive an evidence when the user is interacting in that situation. Figure 6 represents a BN aiming at modelling user preferences and interests when on *Holiday*.

In this prototype, in order to test the suitability of our approach and to simplify the propagation algorithm, we implemented a simple model in which a causal link connecting two macro-nodes represent a relation between a leaf node (origin of the causal link) and a root node (destination of the casual link). The *budget* node, in the BN in Figure 6, is an example of this situation.

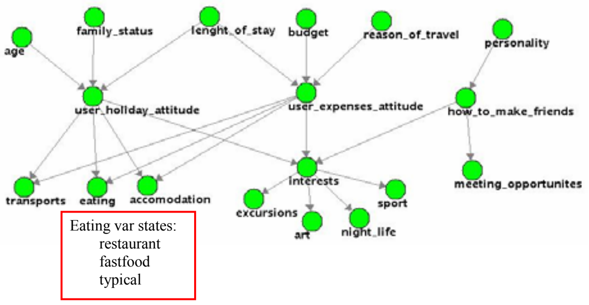


Figure 6: Example of BN for Modeling Holiday Interests and Preferences

When the UMA detects a situation that could require the use a particular portion of user model it transfers the correspondent network on the personal device. In the current version of the system a transfer is activated according to the user to do list [4], or on user request or, in presence of a network connection, it can be done according to the scope of the environment in which the user is interacting and the user task (i.e. required service in our example). For instance, in the current example, the agent could transfer, on the personal device, the portion of the user model concerning "holidays" since the user has this entry in the to-do list or when he/she starts the interaction with the tourist information service.

As far as interaction with an active environment is concerned, when the user approaches one of the active points in the environment, the UMA, using the appropriate user modelling portion, provides information about the user preferences that are relevant for that particular domain.

These preferences are transferred as XML-annotated user info. The environment can then make its own reasoning about these preferences and adapt interaction accordingly. However, information inferred by the agent has to be passed to the environment in a "understandable" way.

A solution to this problem is to make reference to an ontology so that the Environment can give the right semantic interpretation to user data. In our system we transform inferred information in **situational statements** [20]; this language is able to integrate user features and data with situational statements and privacy settings in order to support ubiquitous interaction.

During the interaction, the UMA sends to the environment a XML string, representing situational statements relevant for that domain and interaction context.

A situational statement has the following form:

```
<situation><statement id="1">
<mainpart subject="nadja" auxiliary="HasInterests"
predicate="eating" range="restaurant|fastfood|typical"
object="restaurant" />
<constraints start="17.15.00" duration=""/>
<explanation creator="" method="UMA-BN"
evidence="Equal(reason-of-travel, business) AND
Equal(budget,high)" confidence="0.7"/>
<privacy owner="nadja" access="public"
purpose="information" retention="middle"/>
</statement></situation>
```

This statement expresses the fact that the user prefers to eat at restaurants (*mainpart* tag) with a confidence of 0.7 when she is travelling for business and has an high budget (*explanation* tag).

In order to transform the results of the net into situational statements, or potentially any other language with similar expressive capability, it is necessary to query the variable of interest in the net (i.e. eating), that is the *predicate* of the "mainpart" of the statement, and read all the couple (variable-states, probability) for that variable. The name of the states will become the *range* and the one with significant highest probability will become the value of the *object* attribute of the "mainpart"; the probability value will become the *confidence* of the "explanation" tag. In case of equal probability the UMA considers this as a uncertain belief and then does not state any predicate and confidence value. The *evidence* attribute of the "explanation" tag represent the context in which that fact is true. We can easily represent the context as a conjunction of evidences that bring to that decision. However, we think that from the environment information provision point of view this is not relevant. What is relevant is that "Nadja in the current situation strongly prefers restaurants". Privacy information are set by reading user preferences for that category of information.

User moves in the digital and physical world are detected and used to update the user model. In particular, during the interaction the Environment Agent sends to the Personal Device Agent the APML string representing information tailored to user preferences and the list of all the possible alternatives or related topics at that level of the dialog. In this way the user may acknowledge proposed information or manifest a different interest. These data will be collected by the UMA in a usage model [9] where each action has a weight that will be used to update the user model (probability tables) after a number of interactions (calculated as a significant percentage on the total number of interaction) in that kind of environment and context. This update is reflected in the global HBN tree according to a simple BN propagation algorithm. We are still investigating how this model can be improved in order to deal with more complex network structures.

AN EXAMPLE OF INTERACTION

Let us describe how the system works using as an example the Tourist Information domain. Let's suppose the relevant user model network has been already transferred on the personal device.

As described previously, according to the location and the environment settings the initial environment and agent models are triggered.

If the user interacts with the Environment Agent with his/her personal device, when he/she is in proximity of an information access point, the Environment Agent can be activated and user-related information may be transferred to it. Therefore, information provided may be adapted to that user, who can handle interaction with the Personal Device.

Let's suppose that the Environment Agent, according to its plans, shows to the user the list of available services as shown in Figure 4. Let's suppose the user selects the *transportation* icon, in correspondence of this action the UMA will query the *transport* variable in the BN and will pass to the environment the correspondent value as illustrated in Section 2.2. So, for instance, in case the user is a teen-ager on holiday, the UMA will get as a result that the user prefers public transportation with a probability of 0.6. Then, the Environment Agent will focus its suggestions on this kind of information and, when the user will select the "transportation" icon on the interface, the environment agent, according to its plans, will show public transportation information and will let the user free to select other categories by the apposite icons.

If the user interacts with the ECA displayed in a public kiosk (Figure 3), then she will provide the same information combining verbal and non-verbal expressions according to the generated APML string. In this case, the user may interact using icons on the public touch screen.

CONCLUSIONS

This research builds on prior work of various research teams and of our group on dialog simulation [6] and personalization in ubiquitous computing [4]. In this paper we discussed, in particular, the architecture of a distributed system supporting personalization of interaction in public spaces, where the user brings his/her user model on the personal device.

In particular we adopted and agent-based approach that uses Belief networks for reasoning about user preferences and interest in a given domain (triggered by the environment in which the interaction takes place and the user task).

The interaction may be performed using a public touch screen or a personal device. In the first case we use an ECA to represent the environment. However, we are convinced that interaction through a personal device allows to overcome to limit typical of public space interaction such as lack of personalization in presenting the information and I/O problems due to the noisy nature of the environment. This is a first step in the direction of building true mobile user modeling agents. Our future work will concentrate on refining how the UMA can take advantage of the context-aware in order to decide with portion of the user "global" model is relevant in a particular environment and how the contextual and global models can be updated avoiding inconsistencies and redundancies.

ACKNOWLEDGMENTS

Research described in this paper is an extension of the work we performed in the scope of the European Project Magicster (IST 1999-29078). Besides our project partners, thanks those who cooperated in implementing the prototype described in this paper: in particular, Vincenzo Silvestri and GianLuigi Del Vecchio. Finally, we thank Fiorella de Rosis to which we owe several fruitful ideas underlining this work.

REFERENCES

1. Bickmore T. and Cassell J. Relational Agents: A Model and Implementation of Building User Trust. ACM CHI 2001 Conference Proceedings, Seattle, Washington, 2001.
2. Cassell J., Stocky T., Bickmore T., Gao Y., Nakano Y., Ryokai K., Tversky D., Vaucelle C., Vilhjálmsson, H. (2002). MACK: Media lab Autonomous Conversational Kiosk. Proceedings of Imagina02. Monte Carlo.
3. Cavalluzzi A., De Carolis B., Pizzutilo S., Cozzolongo G.: Interacting with embodied agents in public environments. AVI 2004: 240-243
4. De Carolis B., Pizzutilo S. and Palmisano I. D-Me: Personal interaction in Smart Environments.In P Brusilovsky, A Corbett and F de Rosis (Eds): Proceedings of User Modeling '03.Springer LNAI 2702. 2003.
5. De Carolis B., Pelachaud C., Poggi I. and Steedman M. APML, a Mark-up Language for Believable Behavior Generation. In H Prendinger and M Ishizuka (Eds): "Life-like Characters. Tools, Affective Functions and Applications". Springer, in press.
6. de Rosis F., De Carolis B., Carofiglio V. and Pizzutilo S. Shallow and inner forms of emotional intelligence in advisory dialog simulation. *In H Prendinger and M Ishizuka (Eds): "Life-like Characters. Tools, ffective Functions and Applications"*. Springer, in press.
7. de Rosis F., Pelachaud C. and Poggi I. Transcultural believability in Embodied Agents: a matter of consistent adaptation. In S. Payr and R. Trappl (Eds): "Agent Culture: Designing Human-Agent Interaction in a Multicultural World."Laurence Erlbaum Associates, 2003.
8. Dey A. K. Understanding and Using Context. *Personal and Ubiquitous Computing* 5 (2001) 1, 4-7
9. Fink J. and Kobsa A. (2002): User Modeling in Personalized City Tours. *Artificial Intelligence Review* 18(1), 33-74.
10. Gratch J., Rickel J., Elisabeth Andre, Norman Badler, Justine Cassell, Eric Petajan (2002) Creating Interactive Virtual Humans: Some Assembly Required IEEE Intelligent Systems 17(4): 54-63.
11. Gyftodimos E. and Flach P.A..Hierachical Bayesian Network: A Probabilistic Reasoning Model for Structured Domains. Proceedings of the ICML 2002 Workshop on Development of Representations. pp. 23-30.
12. HAPTEK: http://www.haptek.com
13. http://www.fipa.org.
14. Kobsa A., Generic User Modeling Systems. UMUAI vol. II nos.1-2 pp.49-63. Kluwer Academic Publisher. 2001.
15. Pelachaud C., Poggi I., Subtleties of Facial Expressions in Embodied Agents, Journal of Visualization and Computer Animation, to appear, 2002 .
16. Poggi I., Pelachaud C., de Rosis F., Carofiglio V. and De Carolis B. Greta. A believable embodied conversational agent. In O Stock and M Zancanaro (Eds): "Intelligent Information Presentation". Kluwer Ac Publishers, in press.
17. Poggi I., Pelachaud C., de Rosis F. Eye Communication in a Conversational 3D Synthetic Agent, In E. Andrè, (Ed.), Special Issue of Artificial Intelligence Communications, The European Journal on Artificial Intelligence, IOS Press, vol 13 (3): 169-181, 2000. zipped Postscript or PDF
18. Reiter E. and Dale R. Building Natural Language Generation Systems. Cambridge University Press. 2000.
19. Samulowitz M. Designing a Hierarchy of User Models for Context-Aware Applications. Position Paper at www.teco.edu/chi2000ws/papers/23_samulowitz.pdf
20. UbisWorld: http://www.u2m.org