Unobtrusive Personalized Services in Ambient Media Environments

Estefanía Serral, Miriam Gil, Pedro Valderas, Vicente Pelechano

Centro de Investigación en Métodos de Producción de Software
Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain
Email:{eserral, mgil, pvalderas, pele}@pros.upv.es

Abstract. In the age of ambient media, people are surrounded by lots of physical objects (media objects) for rendering the digital world in the natural environment. These media objects should interact with users in a way that is not disturbing for them. To address this issue, this work presents a design strategy for augmenting the world around us with personalized services capable of adjusting its obtrusiveness level (i.e., the extent to which each service intrudes the user's mind) by using the appropriate media objects for each situation.

Keywords: Unobtrusive ambient media; Personalized services; Context awareness

1 Introduction

Ambient media are embedded throughout the natural environment of users in media objects that surround them (e.g., in his mobile device, in his car, in the TV, in the lights, etc.) [10]. These media objects must behave invisibly and unobtrusively to promote a natural interaction between the user and the environment. Implicit interactions, those that occur without the explicit behest or awareness of the user, become increasingly important to achieve this invisibility and not being obtrusive [9].

This work is focused on supporting unobtrusive personalized services using the media objects throughout the environment. These services will be proactively executed in reaction to implicit interactions based on the situational context such as location, surrounding environment or user's state [14]. For example, when a hotel guest approaches to his/her room (implicit interaction), a welcome hotel service will be automatically executed: opening the door room to let the user come in, switching the lights of the room on, turning on the air conditioner and communicating the new messages/news of the hotel to the user.

We must design ambient media in a way that negotiate the volume and the medium of interactions with their users, pending the user's current needs. Our design strategy to solve this problem is based on making ambient media more considerate [3] and less interruptive. For example, communicating the new messages using the ambient sound may be annoying either when the user is busy or with company. According to the Considerate Computing vision [3], user attention is a primary resource to be considered. Thus, ambient media services must behave in a considerate manner, demanding the proper level of user attention. Considering the example service introduced above, each time it is performed, some of its actions could require a different degree of user attention. As a consequence, different media objects should be used to support each task at the appropriate obtrusiveness level (e.g., using either the TV or the mobile device to show the hotel messages).

The main contribution of this work is a design strategy for describing unobtrusive personalized services in ambient media environments using high-level abstraction models. These models specified at design time are leveraged at runtime for supporting the described behaviour in an unobtrusive way. User needs drive the design of the system providing users with personalized services and avoiding information overload.

The paper is organized as follows. Section 2 describes the high-level abstraction models for describing unobtrusive personalized services. Section 3 explains how these models are used at runtime for providing the functionality of these services. Section 4 describes the related work. Finally, Section 5 discusses the work and presents conclusions and further work.

2 Modelling Unobtrusive Personalized Services

The goal of the design strategy presented in this work is to manage the media objects of the environment to augment the world around users and provide them with unobtrusive personalized services. This interactivity with the physical objects should be adapted to the attentional resources and needs of each user in order to avoid overwhelming the user. We propose personalizing services in ambient media environments by using the most appropriate media object for each situation according to users needs and preferences. For defining such services, designers need to perform the following steps:

1. ***User Modelling***: Detect user needs and preferences to determine the obtrusiveness level required for the interaction with ambient media services.
2. ***Service Modelling***: Define the service behaviour and interaction style based on these needs and obtrusiveness to make use of the appropriate media objects for each situation.
3. ***Media Object Modelling***: Define the implication of this service behaviour and attentional resources for choosing the most suited media objects.

2.1 User Modelling

To detect user needs and preferences for achieving the personalization of ambient media services we make use of **personas** (also known as user profiles). A persona is a summary representation of the system's intended users, often described as real people [1]. Personas provide a framework for describing the target audience in a way useful to design systems. In this way, services can be personalized according to these descriptions.

Personas are used to gather the relevant information of the audience to help drive design and detect common functionalities between users. Personas are usually used in the design of user-centered approaches. According to the users, personas give a much more concrete picture of typical users providing features that directly address specific user needs [4]. Thus, it is interesting the use of them in this work where we have directly address specific user needs.

Personas describe a target user of the system, giving a clear picture of how they are likely to use it, and what they will expect from it [1]. It has become a popular way for design teams to capture relevant information about customers that directly impact on the design process: user goals, scenarios, tasks, and the like. Scenarios are little stories describing how typical user tasks are carried out. They help to identify both the decisions that a user will have to make at each step in their experience and his/her preferred interaction techniques.

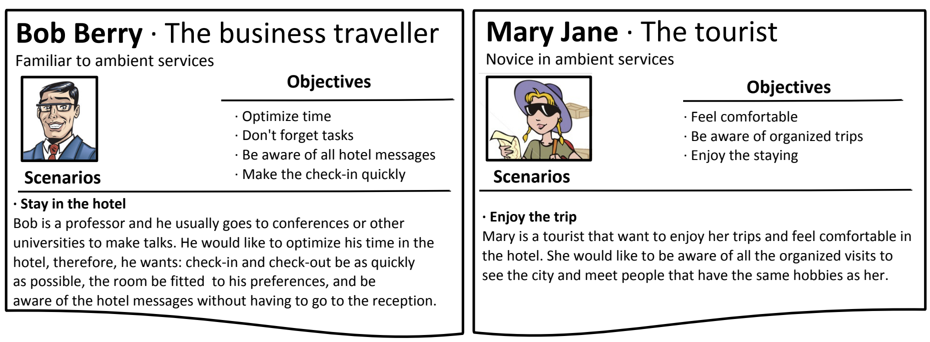

Fig. 1. Excerpt of two personas

In this work, we follow the notation defined by [1] to determine the needs of each user and the functionality required in a specific domain. Figure 1 shows an example of two personas for a Smart Hotel system. These personas give a detailed picture of a business traveller and a tourist. Both have different objectives and have different scenarios according to their needs. For example, the business traveller is a busy person that want optimize his time, be productive and be aware of the hotel messages in all moment. In contrast, the tourist wants to feel comfortable during the staying in the hotel and be aware of the organized trips. Thus, these different personas require different services and different interaction with the services according to their needs. For example, the business traveller would need an ambient service executed when he enters the room in charge of fitting the room to his preferences and remember him the missed hotel messages. However, the tourist would need a service that notifies her the organized trips. In this way, personas will guide subsequent adaptations in information presentation, modality and media objects chosen. By the definition of personas, analysts must detect the services needed for interacting with the system.

2.2 Service Modelling

Once the needs and the functionalities are identified, analysts have to determine the ambient services that are going to give support to these functionalities and their relationships with contextual information. The way to describe these services in an unobtrusive context-sensitive way is explained in the following subsections.

Describing the service behaviour. In order to describe service behaviour we use task models inspired by Hierarchical Task Analysis, which hierarchically refines a high-level task into more specific ones by building a task tree.

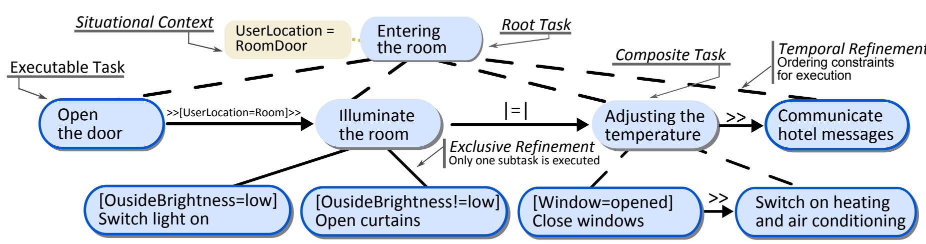

Fig. 2. *Entering the room* Service

Figure 2 shows the modelling of the service behaviour used as example in Section 1 (the *Entering the room* service) using the proposed task model. This service is modelled for the needs of *the business traveller persona*. The root task represents the service and has an associated situational context whose fulfilment starts the execution of the service (*UserLocation=RoomDoor*). This situational context defines the context conditions that are produced in an implicit interaction. The root task is broken down into more specific tasks until they are executable tasks (i.e., tasks that can be executed by a media object). For each executable task, it is indicated in which obtrusiveness levels can be executed, and also, which media objects must be used in each one of those obtrusiveness levels to carry out the task (this is next further explained).

A task can have a context precondition (represented between brackets), which defines the context conditions that must be fulfilled so that a task is performed (if the precondition is not fulfilled, the task will not be executed). For instance, the *Switch light on* task has as precondition that the outside brightness is low.

To break a task down, *temporal* or *exclusive* refinements can be used. On the one hand, using temporal refinements, all the subtasks are executed following a temporal order; in this case, tasks are related between them using temporal restrictions based on the ones proposed in [11]. For instance, as shown in Fig. 2, the service is decomposed into four subtasks using temporal refinements. This means that, when user is detected in front of his/her room hotel, its door will be opened; then, when the user enters inside, the room will be illuminated and the temperature will be adjusted; and finally, the hotel messages will be communicated to the user. On the other hand, using exclusive refinements only one subtask will be executed. For instance, the *illuminate the room* composite task is decomposed into two subtasks by using exclusive refinements; therefore, to illuminate the room either the lights will be switched on or the curtains will be opened. To specify the context conditions (in the situational context, task preconditions and relationships), we use logical expressions grounded on elements of an ontology-based context model (we use the one presented in [16]).

Specifying the obtrusiveness space. To adjust the obtrusiveness of the executable tasks, we make use of the conceptual framework presented in [9] to determine the obtrusiveness level of each task in the system. This framework defines two dimensions to characterize interactions: *initiative* and *attention*. According to the *initiative* factor, interaction can be *reactive* (the user initiates the interaction) or *proactive* (the system takes the initiative). With regard to the *attention* factor, an interaction can take place at the *foreground* (the user is fully conscious of the interaction) or at the *background* (unadvised interaction). For this work, we have divided the attention axis in three segments as shown in Figure 3 which are associated with the following values: *invisible* (there is no way for the user to perceive the interaction), *slightly-appreciable* (usually the user would not perceive it unless he/she makes some effort), and *completely-awareness* (the user becomes aware of the interaction even if he/she is performing other tasks). This conceptual framework defines the space of possible interaction styles (e.g., those that require more or less attention and more or less initiative).

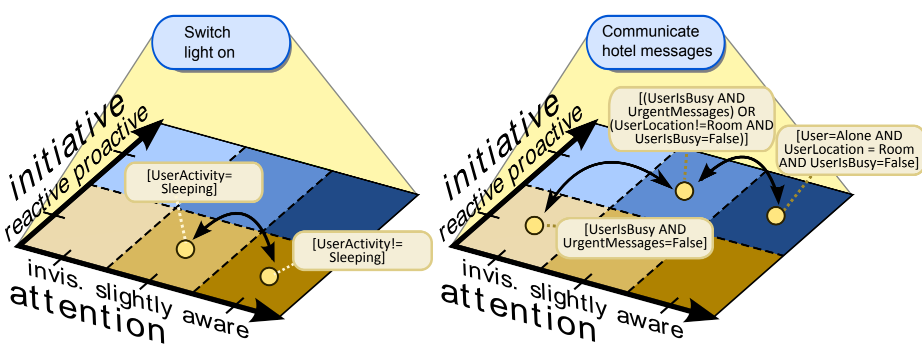

Fig. 3. Selection of the obtrusiveness levels for two tasks

There is different ways to accomplish a task with different degrees of attentional demand and initiative. The selection of one will depend on the user needs and preferences. Furthermore, not all the tasks have sense in all the quadrants. For example, the task *Switch light on* has more sense in reactive levels (e.g., when the user turns on the lights or the lights are turned on when the user enters the room). Designer not only decides what action needs to occur, but also, very importantly, the manner in which it should take place by means of selecting the the possible obtrusiveness levels in which each executable task can be performed. Thus, designers have to describe the obtrusiveness space for each different executable task by defining in which obtrusiveness levels it can be executed and the context conditions that make the system to choose the adequate level each moment. By means of the obtrusiveness space, designers can better match an appropriate interaction to the situation at hand. It is worth noting that the obtrusiveness space is specified for each executable task of each persona, independently of the service to which it belongs.

Figure 3 illustrates the linkage between the executable tasks "Switch light on" and "Communicate hotel messages" (shown in Figure 2) and the obtrusiveness space for the interactions that support these tasks. On the one hand, the task to "switch light on" is performed in different levels of attention depending on the user activity. Note that this task can be executed in other services (such as for waking up the user), and the obtrusiveness space depends only on the task and the persona, not on the service in which the task is performed. Thus, this task can regulate the light using the gradual lighting when the user is sleeping. However, when the user is not sleeping, the task is performed at the completely-awareness level of attention and, therefore, it uses the regular lights. This task is always performed in a *reactive manner* with regard to the initiative axis since the task is executed in reaction to the user detection. On the other hand, the second task provides the hotel messages to the user. Depending on the urgency of the messages, the location of the user and the engagement of the user in other activities and his/her company, the task is executed in a different obtrusiveness level. The system informs the user about the messages *proactively* in a notorious manner requiring a high level of attention (*completely-awareness*) if the user is alone in the room and is not busy (e.g., ambient sound can be used). However, if the user is not in the room and is not busy, or is busy but the messages are important, the system informs the user about these messages in a subtle manner (e.g., using his mobile device). Otherwise, if the user is busy and the messages are not urgent, the task is carried out in an *invisible manner* without explicitly notifying to the user, only sending the messages to the user mobile device.

Once the context conditions of the obtrusiveness levels selected are defined, we map the obtrusiveness levels to the appropriate media object that support the underlying system. This is illustrated in the following subsection.

2.3 Media Objects Modelling

One of the aspects to be considered in ambient media environments are the physical objects available to interact with users. Thus, an aspect to be modeled is the available media objects and which of them are appropriate to demand the right user attention in each obtrusiveness level.

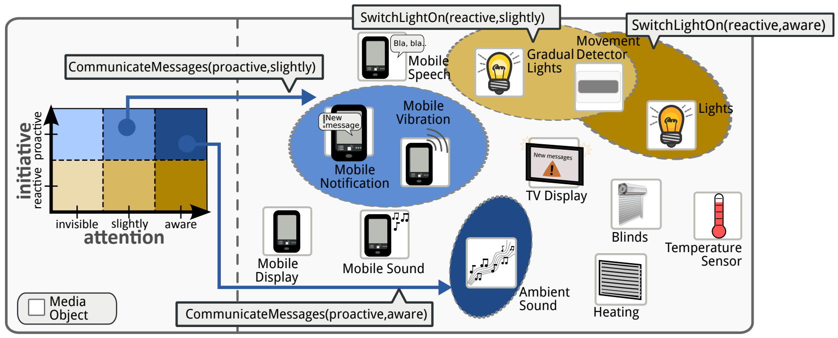

Fig. 4. Implication of the media objects with the attentional demand

To achieve this, designers indicate the media objects that are required for supporting a task in the different obtrusiveness levels in which the task can be performed. Figure 4 shows the part of the media objects that supports some tasks from the example scenario explained in Section 2.2. For example, the *ambient sound* of the room is used for the "communicate hotel messages" task when the task is performed proactively at the highest level of attention axis. This mapping between the *ambient sound* and the obtrusiveness level for the "communicate hotel messages" task is expressed as *CommunicateMessages(proactive,aware)* where the first coordinate refers to the initiative and the second one to the attention axis (see Figure 4). However, the *mobile notification* and the *mobile vibration* are used for the same task when the task is performed at the slightly level of attention. Another example is for the "switch light on" task. The *gradual lights* that regulate the light are used for this task when it is performed at the slightly level of attention. Conversely, the *regular lights* are used when the task is performed at the highest level of attention (i.e., *completely-awareness*). Figure 4 illustrates these mappings for the tasks.

3 Executing Unobtrusive Personalized Services at Runtime

The models proposed in the previous section allow unobtrusive personalized services to be described regardless the particular technology used for the implementation. To achieve the execution of such services, these models are prepared to be interpreted at runtime. Thus, all the efforts invested at design time would be reused at runtime providing new opportunities for adaptation capabilities without increasing development costs. The software infrastructure presented in [15] has shown its feasibility and correct performance to execute complex services by interpreting the presented task model. This infrastructure could be extended for supporting the unobtrusive and personalized execution of the services designed by following the approach presented in this work. Specifically, the extended software infrastructure should perform the following process for executing a designed unobtrusive personalized service by interpreting the models at runtime:

1. *Sensing context*: The media objects deployed in the smart environment sense context changes, which are monitorized and captured in a context model. These media objects are also capable of identify each user. For instance, when Bob is approaching to his room, his location and identification are sensed by RFID and presence detector objects.
2. *Retrieving personalized services*: Once a context change is detected, the services defined for the identified user according to her/his persona are retrieved. For instance, since Bob location has changed, the services described for his persona (*the business traveller*), such as the *Entering the room* service described in Section 2.2, are retrieved.
3. *Checking situational contexts*: The situational context of the retrieved services (e.g., UserLocation=RoomDoor) are checked to see if any of them is satisfied and the corresponding service has to be executed.
4. *Executing personalized services*: If the situational context of a service is satisfied, the service is carried out in a considerate manner. For instance, when Bob location is in front of his room door, the situational context of the *Entering the room* service is satisfied and the service is carried out. To achieve this, the following steps are executed:
   - *Retrieving executable tasks and obtrusiveness:* The different executable tasks of the service are progressively retrieved according to their refinements, the temporal relationships specified among its tasks and the up-to-date context information. After retrieving an executable task, if the precondition is satisfied (considering it is true if the task has not a precondition), the obtrusiveness space described for the task is retrieved.
   - *Adjusting the obtrusiveness level:* The context conditions specified in the obtrusiveness space are checked to know in which obtrusiveness level the task must be executed. For instance, for the *communicate hotel messages* task, when the user is not busy and is alone, the task must be carried out in the *proactive-aware* obtrusiveness level (see Figure 3).
   - *Using the appropriate media objects:* The media objects that must be used to perform the task for that obtrusiveness level are searched for. Given a task and an obtrusiveness level, the mapping defined between media objects and obtrusiveness levels returns the media objects to be used. For instance, for the *communicate hotel messages* task the *ambient sound* is used in the *proactive-aware* obtrusiveness level (see Fig. 4).

4 Related Work

This work supports the designing of unobtrusive personalized services that interact with users by using ambient media objects. Related work on interactive systems has been proposed to deal with modelling system interaction that adapt to users. These approaches enable to exploit the information contained in the models for interaction adaptation to changing contexts. Calvary et al. [2] give an overview of different modelling approaches to deal with user interfaces supporting multiple targets in the field of context-aware computing. The work of Hervás et al. [5] presents a conceptual model to link context information with pervasive elements in order to personalize the offered services of these elements. However, these works define the adaptation space in terms of the environment and the platform. Our approach defines the adaptation space in terms of obtrusiveness for achieving seamless interactions. Thus, we address a different issue that is more related to human limitations (e.g., user attention) than device technical limitations (e.g., screen size).

Towards creating systems that adapt their level of intrusiveness to the context of use, works are mainly focused on minimizing unnecessary interruptions to the user [13]. Moreover, the amount of studies concentrating on the design of unobtrusive interactions is still limited. Approaches in the area of Considerate Computing [3] are mainly focused on inferring attention in order to predict acceptability. Horvitz et al. [8] demonstrated the potential use of Bayesian networks for computing the cost and value of interruptions. Hinckley and Horvitz [6] modelled interruptibility by considering the user's likelihood of response and the previous and current activity. Ho and Intille [7] suggested that proactive messages delivered when the user is transitioning between two activities may be received more positively. Vastenburg et al. [17] conduct a user study of acceptability of notifications to find out what factors influence the acceptability of them. Although these initiatives recognize the need to adapt the interaction, efforts have been put on minimizing unnecessary interruptions, overlooking automation aspects that we have taken into account. Moreover, we propose to represent behaviour aspects in the interaction design level in order to define how the different interactions are adapted when a particular situation is produced.

5 Conclusions and Further Work

In this work, we have presented a design strategy for describing unobtrusive personalized services in an ambient media environment. The provided models describe how services must interact with each Persona in an unobtrusive way. Using these abstract models, the system is designed by using concepts that are much less bound to the underlying implementation technology and are much closer to the problem domain [12]. This makes the models easier to specify, understand, and maintain than code. These models are also defined to be leveraged at runtime to execute the needed services in such a way that they demand the adequate user attention in each situation. Moreover, by handle obtrusiveness as a separate concern, a given task can be presented to each Persona in a complete different manner just by changing the obtrusiveness space specification, without altering the service description. As further work, we plan to validate the presented design strategy and to extend the software infrastructure presented in [15] for supporting the interpretation of the presented models at runtime and the unobtrusive execution of the personalized services.

Acknowledgments. This work has been developed with the support of MICINN under the project EVERYWARE TIN2010-18011 and co-financed with ERDF, in the grants program FPU.

References

1. Dan M. Brown. *Communicating Design: Developing Web Site Documentation for Design and Planning (2nd Edition)*. New Riders Press, 2010.
2. Gaelle Calvary, Joelle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, June 2003.
3. W. Wayt Gibbs. Considerate computing. *Scientific American*, 292(1):54–61, 2005.
4. Jan Gulliksen, Bengt Goransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Asa Cajander. Key principles for user-centred systems design. *Behaviour & Information Technology*, 22:397–409, 2003.
5. Ramón Hervás, José Bravo, and Jesús Fontecha. Awareness marks: adaptive services through user interactions with augmented objects. *Personal Ubiquitous Comput.*, 15:409–418, April 2011.
6. Ken Hinckley and Eric Horvitz. Toward more sensitive mobile phones. In *Proc. of the UIST '01*, pages 191–192, 2001.
7. Joyce Ho and Stephen S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proc. of CHI '05*, pages 909–918. ACM, 2005.
8. Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel. Models of attention in computing and communication: from principles to applications. *Commun. ACM*, 46:52–59, 2003.
9. Wendy Ju and Larry Leifer. The design of implicit interactions: Making interactive systems less obnoxious. *Design Issues*, 24(3):72–84, 2008.
10. Artur Lugmayr, Thomas Risse, Bjoern Stockleben, Kari Laurila, and Juha Kaario. Semantic ambient media an introduction. *Multimedia Tools and Applications*, pages 337–359, 2009.
11. Fabio Paternò. *ConcurTaskTrees: An Engineered Approach to Model-based Design of Interactive Systems*. 2002.
12. Fabio Paternò. From model-based to natural development. *HCI International*, pages 592–596, 2003.
13. Sarvapali D. Ramchurn, Benjamin Deitch, Mark K. Thompson, David C. De Roure, Nicholas R. Jennings, and Michael Luck. Minimising intrusiveness in pervasive computing environments using multi-agent negotiation. *MobiQuitous '04*, pages 364–372, 2004.
14. Albrecht Schmidt. Implicit human computer interaction through context. Technical report, Personal Technologies, 2000.
15. Estefanía Serral, Pedro Valderas, and Vicente Pelechano. Automating routine tasks in ami systems by using models at runtime. In *AmI '10*, pages 1–10, 2010.
16. Estefanía Serral, Pedro Valderas, and Vicente Pelechano. Towards the model driven development of context-aware pervasive systems. *PMC*, 6(2):254–280, 2010.
17. Martijn H. Vastenburg, David V. Keyson, and Huib de Ridder. Considerate home notification systems: a field study of acceptability of notifications in the home. *Personal and Ubiquitous Computing*, 12(8):555–566, 2008.