

# Towards a generic model for user assistance

Blandine Ginon

Université de Lyon, CNRS,  
INSA-Lyon, LIRIS, UMR5205, F-69621, France

**Abstract.** This paper aims to present our generic model for user assistance. The approach we propose allows a designer to describe the assistance requested for an application using a common formalism; this description is then used by a generic assistant. A set of epiphyte assistants is then involved to perform assistance actions specified in the assistance description made by the designer.

**Keywords.** Assistance, user model, formalism, personalization.

## 1. Introduction

Over the years, computer applications have become more and more complex. Facing the numerous functionalities offered by these applications and their often dense interfaces, a novice user may become confused and give up using them despite the rich possibilities they offer. Otherwise, some users may under-exploit these applications, thus limiting their interest and richness. To overcome this difficulty of handling and use, there are two main ways of working: improving the usability, and assisting the user. First, the establishment of a good ergonomics in an application can increase significantly its usability. For this reason, the quality of ergonomics should be a major concern of application designers. However, this does not resolve all the problems encountered by users. For this reason, assuming that the ergonomics problem can be solved, this PhD thesis is focused towards user assistance.

User assistance, also named help, encompasses everything that can allow users to avoid under-exploit an application, or turn to another user, more expert, to help them overcome their difficulties. It is intended for all users (novice, occasional, standards or expert), addressing both first use and everyday use. The development of an assistance system for an application is a complex but important work. However, developers often neglect it to concentrate on the development the application's functionalities.

The main issue of our thesis is to allow designers to create more efficiently an assistance system suited to a given application and fully customizable according to user models. Our aim is to define generic models to describe assistants and applications' expected assistance. We will use an existing generic model to describe users. It will then be possible to define a broker that will use such descriptions to trigger the appropriate assistant for a given application. The description of the application will help the broker in providing context-sensitive assistance. The user model will make it possible to personalize this assistance according to the user' knowledge, abilities, preferences, goals, experiences...

## 2. User assistance

**Typology of user assistance.** For the first step of this thesis, we proposed a typology of user assistance, which presents the user needs (whether novice, occasional or expert standards), and compares the different techniques and approaches of assistance that can meet these needs. Assistance needs of users are varied (discovery functionality, guiding,

improving practice...). Furthermore, these needs vary depending on the application, its complexity and richness, as well as users, their objectives and their experience of the application. Techniques to meet some assistance needs can be as simple as posting messages, or more elaborate, as the use of patterns or models. Assistance approaches implement different techniques, it may be, for example, systems consultants, tutorials or Context help [Erreur ! Source du renvoi introuvable.].

**Assistance model.** In order to propose a generic model of user assistance, we are interested in assistance models, and more particularly in generic models, independent from the application where they will be implemented. The generic model of advisor system proposed by [1] allows to design advisor systems that can be grafted onto an existing application. This approach has been implemented in the Epitalk system, which allows the designer to define, for a given application, a suitable assistance using the advisor system. However, the applications have to be implemented in Smalltalk, and the use of Epitalk requires a good knowledge of Smalltalk and multi-agent programming, which greatly limits its scope. The generic assistance model proposed by [2] aims at actors of distance learning. It consists of a model of the task, a model of the user and the group, and a set of assistance interventions. Assistance interventions can meet the needs of users; it is associated with a set of metadata (user name, goal ...) and a set of rules of the form "If Condition, Then Action". To be reused in the context of our work, the set of proposed actions (change the interface, display messages, change the user model ...) must be enriched to meet the diverse needs of users.

**User Model.** To allow the personalization of the assistance, we must have information about the users. This information may include knowledge (such as a skill level in using the application), their abilities and possible disabilities (such as a hearing or view rate), and their preferences (such as the mode of assistance they prefer). All this information can be collected, deduced or provided directly by the users themselves. It constitutes a *user model*. To exploit this information about a user, it is necessary that they be expressed in the same formalism. We chose to use the profile modeling language PMDLe [3; 4]. It allows to describe the structure of our user model, *i.e.* the way data are organized in the model, and the data itself. In PMDLe, the structure may be common to several users, for example to all users of an application, while the data is specific to each user. PMDLe is already implemented in an environment able to manage (create, populate and edit) user models [4].

**Personalization of the assistance.** To obtain assistance that is effective and accepted by the user, it can be useful to personalize it to suit the context and specificities of the user. Thus, we will not offer the same assistance to a novice user and an expert user, and the proposed assistance will be different for a basic or advanced functionality of the application. An approach to take into account the specificities of users (knowledge, skills, preferences...) consists in defining constraints on the user model. The profile modeling language PMDLe is associated with a model of constraints on profiles cPMDLe [5]. Currently, cPMDLe is implemented for customizing learning activities [6; 5]. Used in another context, cPMDLe can be used to personalize any kind of application using PMDLe for describing user models. For example cPMDLe allows to select users having a knowledge level "novice" for the application, or those whose hearing is between 60% and 80%.

**Epiphyte systems.** An epiphyte system is a system that is grafted on another system without interfering with its operation [1]. An assistant system can be grafted to an application, to provide assistance to the users of this application. Several existing systems provide epiphyte assistance [7]. However, these advisor systems are not generic; they can only be applied to their intended applications. However, some studies propose assistance

both epiphytic and generic, like the Microsoft companions [8] and the animated conversational agents of WebLéa [9] and Cantoche [10].

### 3. Our approach of assistance to user

In this section, we present our generic model of user assistance. The principle of our approach is to allow in a first step a designer to describe the assistance requested for a given application, and in a second step to provide that assistance, as specified by that description, but also personalized to each user. Our approach involves a set of epiphytes assistants (cf. (A) Fig. 1), such as advisor systems or conversational agents. Thereafter, new assistants may be added to this set. To enable the use of these epiphytes assistants, they must be described using a common formalism, an assistant description language (cf. (B) Fig. 1). This description will later allow our generic assistant to drive these assistants and to graft them onto the target application. This description is made only once by an expert each time a new assistant is added to the system.

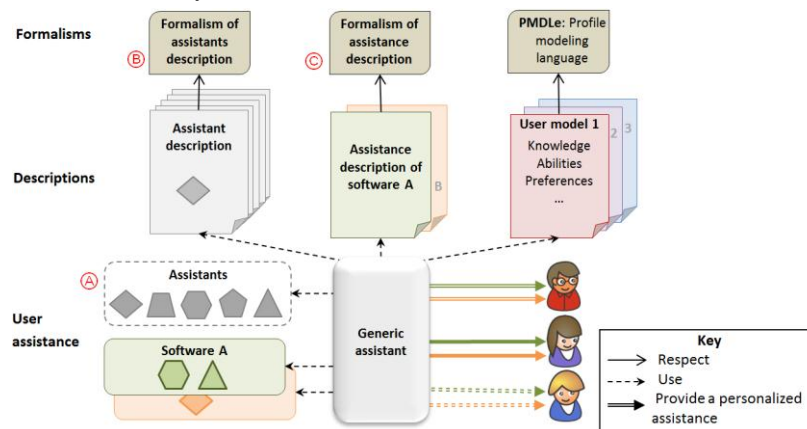


Fig. 1 – Generic assistant model

To set up assistance for a given application, the designer should describe the requested assistance, using a common assistance description language, cf. (C) Fig. 1. Let's take the example of an application with an integrated advisor system and to which a designer wants to add a more complete assistance system, combining several assistance techniques and able to provide each user with personalized assistance. In the description of the requested assistance, the designer can specify for example that a tutorial and a conversational agent must be added to the application. It may also specify that the tutorial is recommended for novice users and available upon request to other users. The choice of the avatar of the conversational agent may depend on the user's preferences. Regarding the advisor system already integrated in the application, the designer can personalize it by specifying for example that the advisor system will frequently intervene to give simple advices on the overall functioning of the application for novice users, while it will intervene less often with expert users, but with an emphasis on very specific advices related to the current task.

The description of the assistance for an application specifies on one hand the assistance techniques that the designer wishes to implement (in addition to any assistance techniques previously integrated to the application) and on the other hand the personalization required for the assistance. To that end, the designer creates assistance rules of the form "If Condition, then Action". In the specification of conditions, the designer can combine conditions on the user model (such as "If the user is a novice" or "If the user is blind"), with conditions on the context (such as "If the user has completed Stage X" or "If the user has clicked button Y"). In actions, the designer specifies which assistant will perform the action: so this action must respect the assistant description *i.e.*, be a possible action of this

particular assistant. For example, if the assistant is a conversational agent whose description states that an action must contain a message and a set of animations (from a list of possible animations), then a possible action could be {message = "Now you can proceed to the next step", animation = "point the finger at the button Next", animation = "smile"}.

Once the assistance designer of an application has specified the description of the requested assistance, our generic assistant will assist each user accordingly. For this purpose, the generic assistant uses together the description of each assistant, the description of the requested assistance, and the user model to control epiphyte assistants when their action is required, but also to personalize any assistance system already integrated into the application as long as it is described using the common assistant description language.

#### 4. Conclusion and perspectives

With the increasing complexity of applications, a need to assist users of such applications has emerged, because of the wide possibilities they offer.

We presented our generic model for user assistance. This model relies on three description languages, used for describing the available assistants, the assistance requested for the application, and the user model. Then, these descriptions will be used by a generic assistant that will provide the required assistance to each user of the application, in a fully personalized manner according to the designer's choices. For that purpose, the generic assistant can appeal to different epiphyte assistants. We are now defining the formalisms for the description of assistance and for the description of epiphyte assistants. We will then implement these two formalisms in an editor aimed at designers. This editor must enable designers to express their choices but they must also be easy to use and help the designers in their task. We will finally implement a generic assistant, able to provide personalized assistance to user of any application, using the assistance description of this application. We also plan to experiment our propositions, by providing assistance to users of different applications. We will make sure that the editor we provide for designers allows them to express their choices without being too complicated. We will also make sure that the assistance provided to end-users of these applications is a real asset for them.

#### References

1. Paquette, G., Pachet, F., Giroux, S., Girard, J.: EpiTalk, a generic tool for the development of advisor systems. In: IJAIED, 7, 349-370 (1996)
2. Dufresne, A., Basque, J., Paquette, G., Léonard, M., Lundgren-Cayrol, K., PromTep, S.: Vers un modèle conceptuel générique de système d'assistance pour le téléapprentissage. In: STICEF, 10, 57-88 (2003)
3. Eyssautier-Bavay, C., Jean-Daubias, S.: PMDL: a modeling language to harmonize heterogeneous learners profiles. In: ED-MEDIA, Lisbon, Portugal (2011)
4. Ginon, B., Jean-Daubias, S., Lefevre, M.: Evolutive learners profiles. In: ED-MEDIA Lisbon, Portugal (2011)
5. Ginon, B., Jean-Daubias, S.: Models and tools to personalize activities on learners profiles. In: ED-MEDIA Lisbon, Portugal (2011)
6. Lefevre, M., Cordier, A., Jean-Daubias, S., Guin, N.: A Teacher-dedicated Tool Supporting Personalization of Activities. In: ED-MEDIA, Honolulu, Hawaii, 1136-1141 (2009)
7. Kosba, E., Dimitrova, V., Boyle, R.: Using student and group models to support teachers in web-based distance education. In: User Modeling, Edinburgh, UK, 124-133 (2005)
8. Microsoft, MSAgents, <http://www.microsoft.com/products/msagent/main.aspx>, (2012)
9. Martin, J.-C., Sansonnet, J.-P., WebLéa, <http://perso.limsi.fr/jps/online/weblea/leexamples/leawebsite>, (2012)
10. Cantoche, Living Actor, <http://www.cantoche.com/index.php?lang=en>, (2012)