Setup of epiphytic assistance systems with SEPIA

Blandine Ginon, Stéphanie Jean-Daubias, Pierre-Antoine Champin, Marie Lefevre

▶ **To cite this version:**

Blandine Ginon, Stéphanie Jean-Daubias, Pierre-Antoine Champin, Marie Lefevre. Setup of epiphytic assistance systems with SEPIA. EKAW - 19th International Conference on Knowledge Engineering and Knowledge Management, Nov 2014, Linkoping, Sweden. pp.1-4. hal-01301121

HAL Id: hal-01301121
https://hal.science/hal-01301121

Submitted on 2 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Setup of epiphytic assistance systems with SEPIA

Blandine Ginon[1, 2], Stéphanie Jean-Daubias[1, 3], Pierre-Antoine Champin[1, 3] and Marie Lefevre[1, 3]

[1] Université de Lyon, CNRS,
[2] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[3] Université Lyon 1, LIRIS, UMR5205, F-69622, France
{name}.{surname}@liris.cnrs.fr

Abstract. We propose a graphical language that enables the specification of assistance systems for a given application, by means of a set of rules. We implemented it through the SEPIA environment, mainly composed of an assistance editor aimed at assistance designers, and a generic assistance engine able to execute the specified assistance for the target-application end-users, without a need to modify this application. SEPIA has been tested out with 71 assistance designers and more than 250 end-users.

Keywords: Language, user assistance, rule-based system, epiphytic tools.

1 Introduction

User assistance is a solution to overcome the difficulties of handling and using software applications. However, the development of an assistance system suitable is a complex and expensive task, often neglected by applications designers. An alternative to the classical approach of development of an assistance system integrated into the application consists in adopting an epiphytic approach to make possible the *a posteriori* specification and execution of an epiphytic assistance system in an existing application. An epiphytic assistance is an assistant able to perform actions in an external target-application, without disturbing its functioning and without a need to modify it [3]. We proposed an adjunction process of epi-assistance system in a target-application in two phases. The first phase involves an assistance designer: an expert of the target-application who wishes to design an assistance system. It enables the assistance designer to describe the target-application and then to specify the assistance that he/she wishes, by defining a set of assistance rules. The second phase involves the target-application end-users. It consists in the execution of the assistance specified by the designer in the target application. This phase manages the knowledge acquired during the assistance specification phase and relative to the target-application and to the assistance required.

In this paper, we present the SEPIA environment that implements this process and makes possible the adjunction of assistance systems in many applications, without programming work for the assistance designer.

2 The aLDEAS language

We propose aLDEAS (a Language to Define Epi-Assistance Systems), a graphical rule language aimed at assistance designers for defining assistance systems. We choose to propose a graphical language, very suitable to the representation by a set of simple rules of assistance systems, in particular epiphytic assistance systems. Its different components (like elementary assistance actions, waits and consultations) can be combined to create assistance actions addressing assistance needs; these rules will then be executed by SEPIA on top of the target-application. An aLDEAS block is a labelled direct graph, it must have exactly one source, which must be a "block start", and one or several sinks, which must all be "block end".

Fig. 1 gives the example of two assistance rules created for the target-application PhotoScape, a freeware for picture correction. Rule R0 contains a wait (the assistance launch, represented by a) and a consultation of the user asking if he/she wishes help (represented by a). R1 (represented by a) will be triggered if the user chooses the option "yes I want help". The rule R7 is triggered by a click on component 228 (the "tools" menu of PhotoScape). R7 triggers an assistance action made up of two elementary assistance actions (represented by a): the enhancement of component 210 (the "red eyes" button of PhotoScape) that will attract the end-user's attention in order to guide him, and a message suggesting the user to click on this button. The component indicates that the click on the "red eyes" button will put an end to the rule R7 and thus delete both the enhancement and the message.

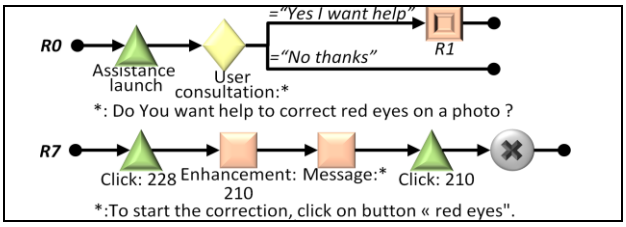

Fig. 1. Example of the assistance rules R0 and R7 for PhotoScape

3 SEPIA ENVIRONMENT

We implemented aLDEAS in the SEPIA environment (Specification and Execution of Personalized Intelligent Assistance) [1]. SEPIA is mainly composed of an assistance editor for the assistance specification, and a generic assistance engine for the assistance execution.

The **assistance editor** is a tool aimed at assistance designers. It implements aLDEAS and makes possible the specification, for any given target-application, of an assistance system described by a set of assistance rules. The assistance editor provides

[1] Demonstration videos of both the specification and the execution of assistance systems with SEPIA are available at http://liris.cnrs.fr/blandine.ginon/PhDWork.html.

the assistance designer with an interface to handle the aLDEAS element in order to define the assistance system. No knowledge on aLDEAS is required to use the assistance editor.

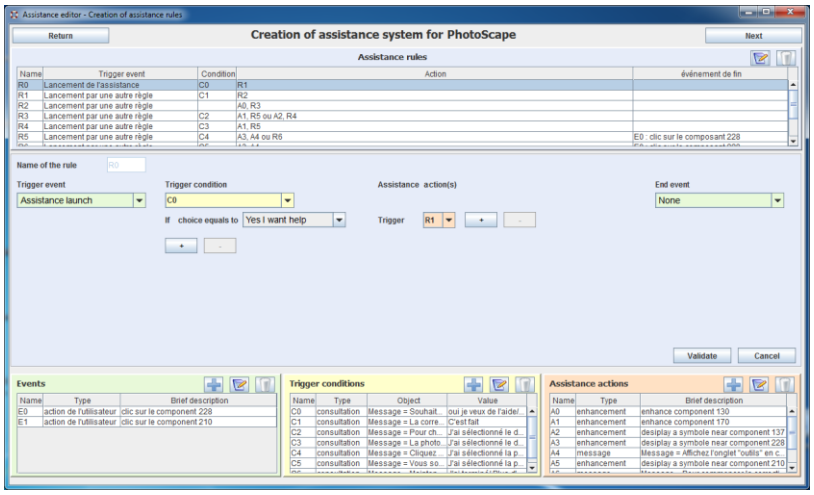

Fig. 2. Screenshot of SEPIA assistance editor

As an example, Fig. 3 shows two zooms on screenshots of SEPIA assistance editor: it represents the creation of rule R0 on top and rule R7 on bottom (cf. Fig. 1).

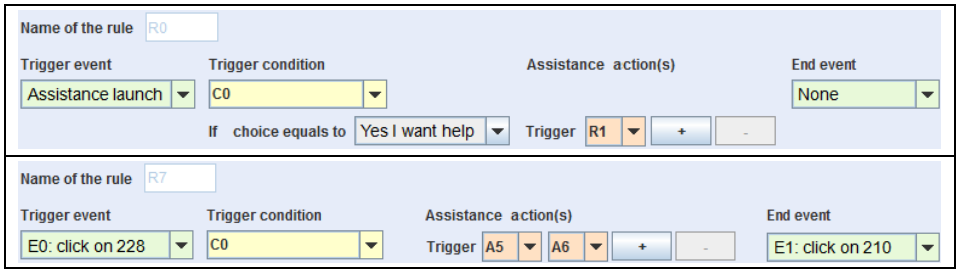

Fig. 3. Zoom on the creation of rules R0 and R7 with SEPIA assistance editor

The SEPIA **generic assistant engine** is then able to execute the assistance specified by the assistance designer in the editor, in order to provide the target-application end-users with personalized and contextualized assistance. To be able to provide contextualized assistance, the assistance engine must be aware of all the end-users actions in the target-application. For example, to execute rule R7 (cf. Fig. 1) in PhotoScape, that must be triggered and end by a click, the assistance engine must be informed when the end-user click on a component of the target-application, and more precisely on which component. For this purpose, we developed a set of detectors [1], based on accessibility libraries and able to monitor nearly any existing target-application: any web applications (except in flash), any Windows native applications, any Java applications for Windows and any GTK or Qt applications for Linux.

To be able to take into account the end-users' preferences, the generic assistance engine also provides end-users with an interface to edit their preference profile. These preferences will be used for the execution of assistance actions for which the assistance designer didn't specify all the optional parameters. As an example, Fig. 4 shows the execution of the rule R7 (cf. Fig. 1) in PhotoScape, for two different end-users: we can observe that the end-users preferences are taken into account for the enhancement (a blue arrow on left and an orange surround on right) and for the message layout.

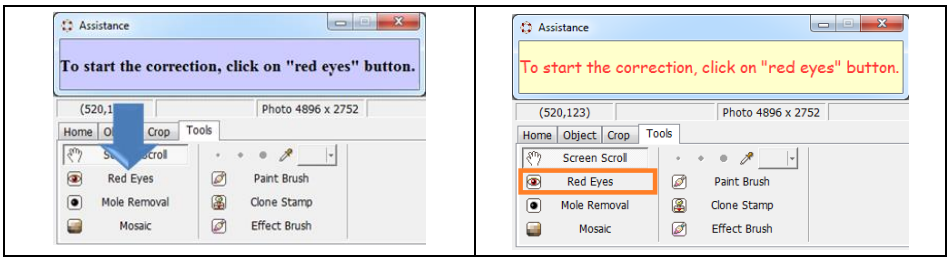

Fig. 4. Personalized execution of rule R7 in the target-application PhotoScape, for two different end-users

4 CONCLUSION AND PERSPECTIVES

The aLDEAS language and the complex assistance action patterns that we proposed allow to define assistance systems suitable to their target-application. By adopting a fully epiphytic approach, we make possible the *a posteriori* plugging of assistance systems in existing applications. These applications have not to be specifically designed for the integration of assistance, no access to its source code is necessary and no programming skills are required of the assistance designer. The aLDEAS language is implemented in the SEPIA system. The experimentations that we performed showed that SEPIA can be used by assistance designers, with or without programming skills. What's more, SEPIA can be used to provide end-users with efficient assistance, in order to help them to achieve a given task, in particular in the context of discovery or occasional use of a target-application.

References

1. Ginon, B., Champin, P.-A., Jean-Daubias, S.: Collecting fine-grained use traces in any application without modifying it. In: Workshop EXPPORT of ICCBR, New-York, USA (2013).
2. Ginon, B., Thai, L. V., Jean-Daubias, S., Lefevre, M., Champin, P.-A.: Adding epiphytic assistance systems in learning applications using the SEPIA system. In: Ec-Tel, Graz, Austria (2014).
3. Paquette G., Pachet F., Giroux S. et Girard J., EpiTalk, a generic tool for the development of advisor systems, International Journal of Artificial Intelligence in Education, vol. 7, pp. 349-370, 1996.