

Using Annotated Task Models for Accessibility Evaluation

Ivo Malý, Jiří Bittner, and Pavel Slavík

Department of Computer Graphics and Interaction
Faculty of Electrical Engineering, Czech Technical University in Prague
{maly,bittner,slavik}@fel.cvut.cz

Abstract. Evaluation of application accessibility is a challenging task that requires an intensive testing with potential application users. An alternative to user tests is the model based testing using simulations. The simulations provide important feedback about application accessibility particularly when it is hard to involve the target users in the tests which is often the case for users with disabilities. In this paper we propose a methodology of providing the quickly and easily necessary data for the simulations. In particular we show how to annotate task models using application walkthroughs logs that is data obtained by recording the application usage. We create annotated task models, which together with the user models are suitable for simulation of application usage by virtual users with various disabilities. We present tools for recording and processing of the application walkthrough logs and tools for the interactive task model annotation. Finally, we provide actual examples of task model annotation on three scenarios involving the Second Life metaverse.

Keywords: Task Models, Accessibility Evaluation, User Centred Design and User Involvement.

1 Introduction and Motivation

The goal of accessibility testing is to gather specific information on accessibility issues. Traditionally the application testing is performed with well selected target group of users. Specifically when dealing with users with disability it may be very difficult to gather the users for testing and the user based evaluation can become very time consuming and expensive. An alternative to the user based evaluation are expert reviews, which however need not discover some issues of the application for the particular user group. An approach that is expected to receive increasing attention in the future is the model based testing, i.e. application testing based on simulations in which we simulate the usage of the application by means of virtual users [1]. The model based testing can be used practically in all stages of application development (from the design to evaluation). As this approach does not employ real users, we can perform significantly more tests to iteratively improve accessibility of the application. Additionally we can reflect

many types of disabilities and also combinations of disabilities for which real users would be hard to reach and test.

Model based testing methods require detailed specification of the application interface in order to perform the accessibility testing. In other words the task model representing the application needs to be annotated with the actual data values which correspond to particular actions expected to be simulated by the virtual users. These data include for example a specification of a screen area in which a mouse click is expected in a particular simulation step, the key to be pressed and the associated time constraints given by the application. Such information is readily available when evaluating web-based applications since it can be interpreted from the standardized user interface descriptions (such as HTML). When dealing with general applications for which a description of user interface is not available (e.g. third party desktop applications, computer games, etc.) it is not clear how to obtain the actual data representing the user interface for performing the simulations.

In this paper we propose a method which allows to specify the required values for the simulation by recording the interaction with the application, annotating the records and binding the records to the task models. The application records are stored in application walkthrough log files combined with a video stream. These logs represent the expected interaction with the application in the particular task. As the task models are independent of concrete layout and size of user interface they can remain mostly static even if the design of the application interface is updated. Such an approach allows to use highly generic task models that can be reused and we only need to annotate the task model using application walkthrough log for a specific version of the application interface.

2 Related Work

Our work relates to the model based testing of GUI applications. Generally, model based testing works with task models representing tasks and with a model representing the application GUI. Aho et. al [5] use custom GUI driver to extract GUI element representation from the application while walking through the tasks in the tested application. This method is limited to Java based applications and no formal description of tasks is used. The GUITAR project [6] is capable of extraction of GUI element representation from the Windows, Java, Android and iOS applications which exploit the standard widgets of the operating system. Again, the tasks are not defined by means of a formal description.

Bertolini et. al. [7] focus on representation of tasks in formal description and their update for the case of user interface modifications. They use CNL (Controlled Natural Language) notation, which is specific subset of English, for description of the task models (called test cases). These task models are then translated into particular script actions that will perform the interaction with the applications. However, it is not clear how the script derive information about GUI elements. Kaklanis [4] uses UsiXML for description of the task models (called simulation models). Again, there is no connection between the model and the information about particular GUI element.