gestUI tool: A Tool to Include Gesture-based Interaction in User Interfaces Through Model-Driven

Otto Parra[1, 2], Sergio España[3], Jose Ignacio Panach[4], Oscar Pastor[2], Verónica Burriel[2]

[1] Computer Science Department, Universidad de Cuenca, Ecuador
[2] PROS Research Centre, Universitat Politecnica de Valencia, Spain
[3] Department of Information and Computing Sciences, Utrecht University. The Netherlands
[4] Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universidad de Valencia,

otpargon@posgrado.upv.es, s.espana@uu.nl, joigpana@uv.es, opastor@dsic.upv.es, vburriel@pros.upv.es

Abstract. gestUI is a software tool that implements a model-driven method to define custom gestures and to include gesture-based interaction in existing user interfaces for desktop-computing. The tool allows software engineers and end-users defining custom gestures in two stages of the software development life cycle: at design stage and at runtime. These gestures are assigned to existing actions (commands) in user interfaces. The proposal is independent of platform and programming language.

Keywords: model-driven method, gesture catalogue metamodel, gesture-based interaction, custom gesture.

1 Introduction

Two factors have contributed to the wide diffusion of computing devices in the most recent years: (a) an appropriate human-computer interaction (HCI), which results in the ease of use of services and software systems available for these devices; and (b) the availability of a wide variety of services, software systems and development tools. Regarding HCI, new types of interactions have been developed, specifically gesture-based interaction. These new interactions are incorporated in user interfaces mainly considering predefined gestures such as tap, double tap, swipe, etc. However, the inclusion of custom gestures in software systems is not easy for developers. They must have knowledge of gestures definition and knowledge of the source code that implements such gestures in different platforms. Building software systems with gesture-based interaction is complicated [1] due to the diversity of devices, software platforms, and development tools to design and implement user interface with gesture-based interaction. In this context, developers require additional knowledge and experience in: (i) the specification and implementation of custom gestures, (ii) the design and implementation of gesture-based user interfaces, (iii) the use of software tools that depend on the target platform to implement gesture-based user interfaces.

In a field where there is a diversity of tools and platforms, and technology evolves rapidly, a model-driven methodology is a valid option to deal with all this variability. Model-Driven Development (MDD) has been fairly popular in the academic community [2] in recent years, and a number of different proposals have been introduced to develop software systems. MDD is a software development paradigm which is based on models that abstractly represent system features. These models are the input for a set of transformation rules that can generate the code that implements such features automatically.

This paper introduces an MDD method to deal with gesture-based user interfaces definition and a tool that supports it, called *gestUI*. The method aims to abstract the definition of custom gestures in such a way that software engineers can focus on specifying features of the gestures through models instead of spending effort on programming language issues. Coding and portability efforts are automated through model-to-text (M2T) transformations. The main contribution is gestUI's ability to reduce the complexity of the process of including gesture-based interaction by automatically generating user interface source code.

We demonstrate gestUI in the context of a desktop-computing software development project. The demonstration illustrates how gestUI helps software engineers to define human-computer interaction based on gestures. We show how to define custom gestures and how to include gesture-based interaction in user interfaces using the tool.

In Section 2 we review the conceptual model of gestUI, Section 3 describes the demo of gestUI and Section 4 covers the final remarks of this paper.

2 Conceptual Model

gestUI [3] focuses on building conceptual models to define platform-independent gesture catalogue and to use model transformations that generates a platform-specific gesture catalogue. This catalogue is used to relate existing actions in the source code of the user interface with gestures that trigger those actions. Additionally, applying M2T transformations, we obtain a new version of the user interface supporting gesture-based interaction.

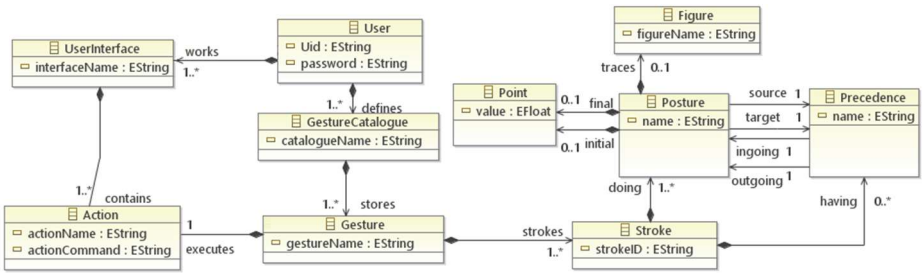

Fig. 1. Metamodel of gesture catalogue

Fig. 1 defines a metamodel that allows representing any custom gestures that users can draw with any system. The metamodel is used in our proposal to save all the supported gestures. A *user* works with one or more *user interfaces*. Each user interface is designed to perform some *actions* (commands) that execute activities/tasks. A user

defines a *gesture catalogue* containing *gestures* to execute these aforementioned actions. Each gesture is composed of one or more *strokes* containing information about the structure of each gesture. *Posture* specifies the orientation of each stroke, each posture has a *precedence* with other, *and Figure* describes the shapes defined by each *point* that contains the coordinates X and Y of each point of the gesture.

3 Demo of gestUI Tool

Next, we show a demo of gestUI to define custom gestures and to include gesture-based interaction in user interfaces of software systems. gestUI has been developed using Java programming language and Eclipse Modelling Framework, which can be run in Windows, Mac OS and Linux platforms with software systems based on Java and SWT.

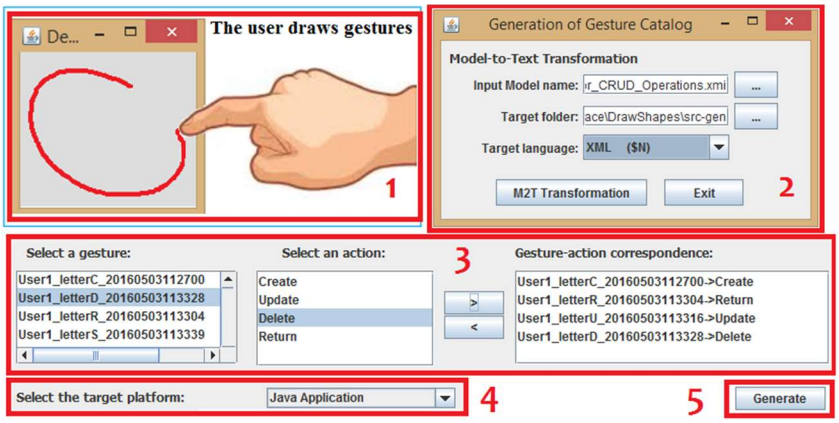

Fig. 2. Sequence of steps to apply gestUI tool

As shown in **Fig. 2**, a user defines custom gestures by sketching on a user interface of gestUI (step 1). The information of the gesture is captured and it is inserted in a model according to the metamodel specified in Fig. 1 [3]. Next, the user applies a model transformation to obtain the gesture catalogue (step 2). Then, the user selects a user interface source code to analyse it and to obtain the actions included in the original source code. Gestures and actions are shown in gestUI with the aim of defining the gesture-action correspondence (step 3). Finally, when the user selects the target platform (step 4), an automatic generation of source code is executed (step 5) to obtain a new version of the source code containing gesture-based interaction.

All this process is run in the demonstration to include gestures to perform actions defined in the user interfaces based on SWT in Java (Fig. 3). This system has three options in the main interface to manage data of a company. We consider the first option ("Departments") that employs the typical actions to manage data (CRUD operations): Create, Read, Update, and Delete. We defined a catalogue with five gestures to execute actions: "C", to create a new department; "R", to read the data of a department; "U", to update the data of a department; "D", to delete the data of a department

and “S”, to save the data in a database. Then, we generate a new version of the user interfaces source code supporting gestures.

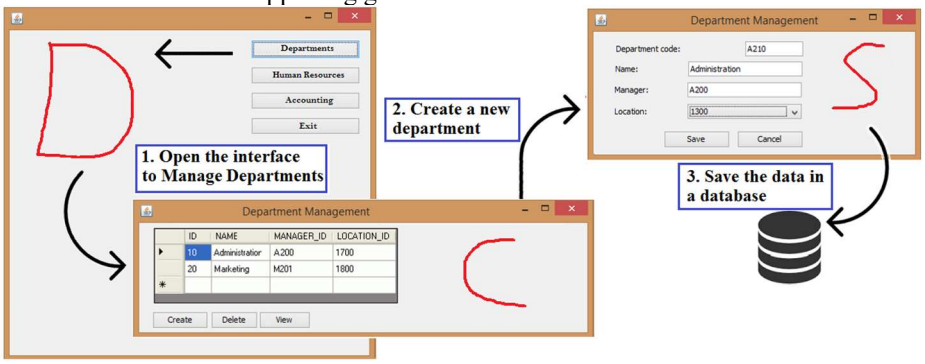

Fig. 3. Software system with gesture-based interaction

4 Final Remarks

gestUI is part of our current research work[1] focused on the inclusion of gesture-based interaction in software systems. Through the model-driven paradigm, we reduce the complexity of the process, helping software engineers in their work of building user interfaces. As result, software engineers define gestures at design stage according to preferences of the users, however, users can modify at runtime, a gesture catalogue according to their preferences. This feature helps in the process of customization of user interfaces where the users can adapt them to their preferences. Further details about our model, tool, and their evaluation can be found at https://gestui.wordpress.com/gestui/.

References

1. Kammer, D. et al.: Towards a Formalization of Multi-touch Gestures. ACM Int. Conf. Interact. Tabletops Surfaces. 3/94, 49–58 (2010).
2. Papotti, P.E. et al.: A quantitative analysis of model-driven code generation through software experimentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7908 LNCS, 321–337 (2013).
3. Parra, O. et al.: GestUI : A Model-driven Method and Tool for Including Gesture-based Interaction in User Interfaces. Complex Syst. Informatics Model. Q. 6, 73–92 (2016).

[1] This work was developed with the support of Generalitat Valenciana-funded IDEO project (PROMETEOII/2014/039)