3
UISKEI

There is a common belief that to build a good user interface, we must refine the solutions iteratively, testing with users to gather feedback and revise them as many time as possible (Szekely, 1994). This ideal interaction design life cycle encompasses a series of inter-related activities, which are part of a greater iterative process. A simple model of this idea can be found in (Preece, Rogers, & Sharp, 2002) and is displayed in the figure below:

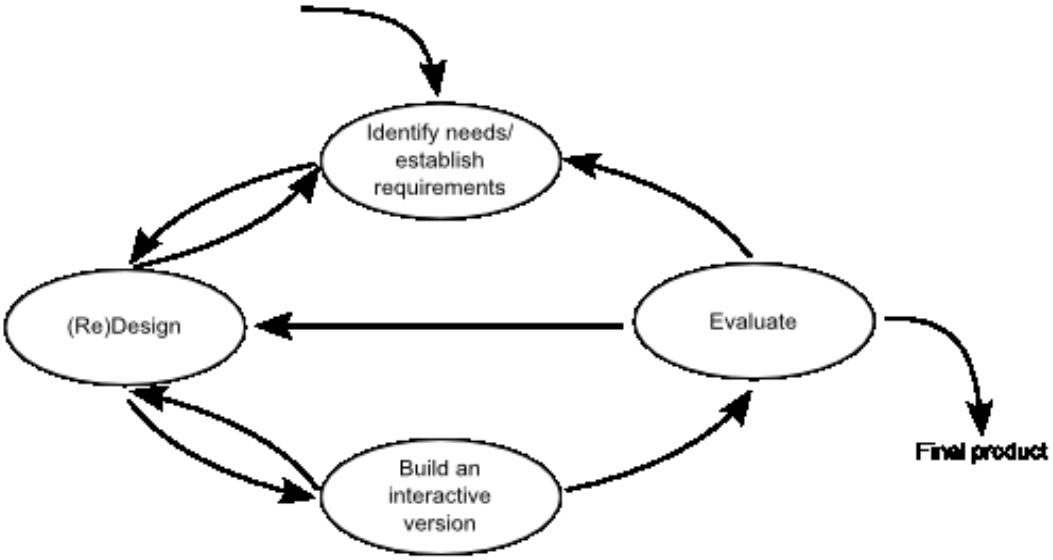

Figure 12: A simple interaction design model[13].

UISKEI (User Interface Sketching and Evaluation Instrument) was developed to aid this life cycle in the early prototyping phase, considering that it can be summarized in three major iterative stages:

1. Interface building → The "(Re)Design" stage, choosing which elements compose the interface and their position, size, etc.;
2. Behavior definition → The "Build an interactive version" stage, describing how the prototype works;
3. Prototype evaluation → The "Evaluate" stage, allowing end users to interact with the prototype to gather their feedback.

[13] Image 6.7, taken from Section 6.4.1 of (Preece, Rogers, & Sharp, 2002)

During the early prototyping phase, sketching can be highly beneficial due to its inherent speed - allowing rapid exploration and iteration of different ideas - and ambiguity - allowing the designer to focus on basic structural issues rather than unimportant details and also allowing multiple interpretations, which can lead to new ideas (Lin, Thomsen, & Landay, 2002). Also, the freeform nature of sketch allows the design to be more creative and exploratory than when using the computer (Hong, Landay, Long, & Mankoff, 2002).

If pen-based interaction resembles the paper experience, why do designers find it easier to sketch on paper? Hammond *et al.* explain this by claiming that "a gulf still exists between the sketch recognition system and the user. To the user, a new mode of interaction is occurring, pen input; however, this conceptual model is inaccurate as the computer still interprets the pen under the mouse/keyboard archetype. No longer can the pen merely stand in for the mouse; rather, a new paradigm of human-computer interaction must be designed around the pen and recognition of the pen input. Pen-based interfaces should provide interpretation and feedback in a natural and intuitive manner, rather than locking the user into mouse-like interactions." (Hammond, Lank, & Adler, 2010)

UISKEI tries to overcome this gulf, creating a rapid "paperless" early prototyping environment, granting the flexibility and speed of the pen and paper version along with interesting computational features, such as moving and resizing.

"Sketching is fundamental to ideation and design. (…) Designers do not draw sketches to externally represent ideas that are already consolidated in their minds. Rather, they draw sketches to try out ideas, usually vague and uncertain ones" (Tohidi, Buxton, Baecker, & Sellen, 2006b). By relying on sketching, we aim to stimulate the exploration of the solution space during the early phases of design. This allows a low cost development of more design alternatives and the possibility to refine them, increasing the chances of obtaining the design right (Tohidi, Buxton, Baecker, & Sellen, 2006a).

Sketching UI designs has already shown advantages according to a study by Kieffer *et al.*(Kieffer, Coyette, & Vanderdonckt, 2010), as follows:

- "UI sketching is preferred over traditional interface builders, especially by end users and could be performed at different levels of fidelity without losing advantages;

- the amount of usability problems discovered with a sketched design is not inferior to those corresponding to a genuine UI;
- the expressive power of a sketched UI remains the same;
- a sketched UI provides quantitative and qualitative results that are comparable to traditional UI prototypes except that the cost is reduced;
- UI sketching encourages exploratory design and fosters communication between stakeholders more than any other prototype;
- flexibility is superior to UI builders, authoring tools, and paper prototypes."

Aiming for the pen-based interaction and to aid in all the three stages of prototyping, the main requirements of UISKEI are detailed below:

- Interface building
  - Produce mock-ups of graphical user interfaces through pen-based interaction;
  - Recognize and convert the sketched elements into interactive elements of the interface model (widgets);
  - Manipulate and edit the widgets
- Behavior definition
  - Define a case, composed by a set of conditions and actions associated to an event;
- Prototype evaluation
  - Evaluate an interactive version of the prototype, in order to realize formative evaluation during the design process;

UISKEI's early version will be discussed in Section 3.1 and the main focus of the new version will be discussed in Section 3.2. The new version approach to each stage will be detailed in further chapters: the interface building will be described in Chapter 4, the interaction definition in Chapter 5, and the prototype simulation in Chapter 6.

3.1
Early version study

A previous version of UISKEI was developed in 2008 as an undergraduate final project (Segura & Barbosa, 2008). This version already explored the drawing of elements which are the basis for interface building, but the behavior definition heavily relied on the form-based interaction, as can be seen in Figure 13.

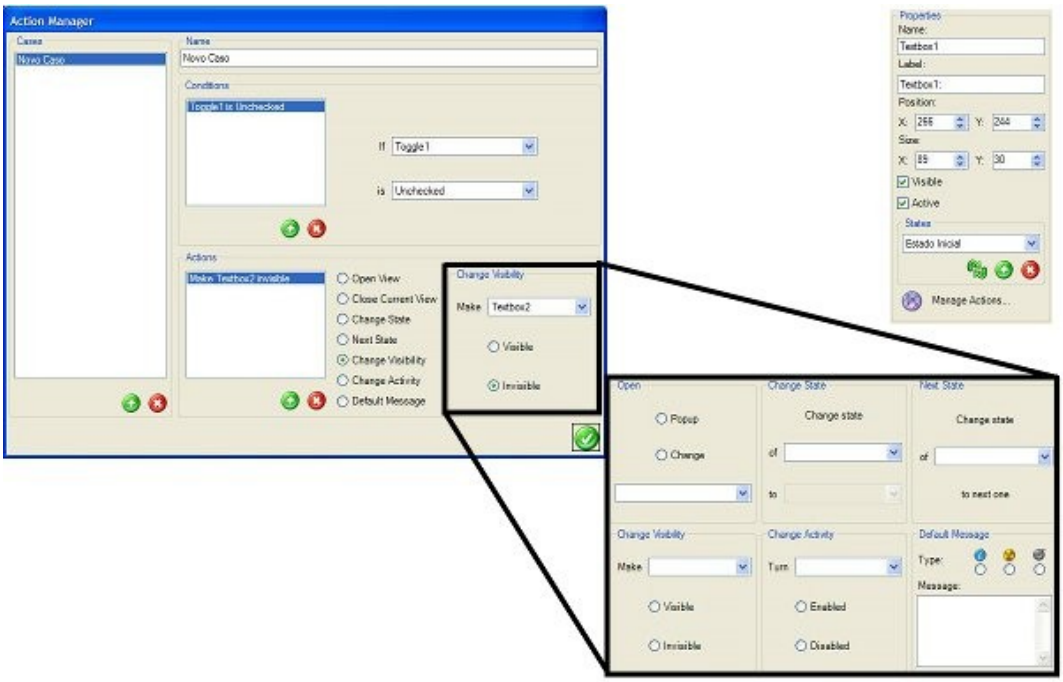

Figure 13: Action Manager of UISKEI's early version.

To add a new behavior, the user should first select the element associated with the event, then click in the "Manage Actions…" button (the lower button on the element's properties pane, pictured in the top-right portion of Figure 13) to open the "Action Manager" window. In this window (pictured in the left portion of Figure 13), the user should add a new behavior case, by clicking in the "Add case" button and then define the conditions and actions associated to the behavior case.

To add a condition, he/she should click in the "Add condition" button and then choose the parameters in two drop-down lists. To add an action, he/she should first click in the "Add action" button, then choose one action type by selecting it amongst the available radio buttons. When the action type was chosen, the parameter pane changed accordingly, as can be seen in the zoomed region of Figure 13, which shows all the possible panes. With this description, it is possible

to notice the amount of mouse clicks needed to define a behavior, completely breaking the pen-based interaction paradigm of the interface building phase.

In addition, the recognition algorithm was hard-coded, restricting the known shapes, and with a recognition rate around 58,6%. An early study (Segura & Barbosa, 2009) considering the end user role and comparing the paper prototyping evaluation technique to an interaction session supported by UISKEI showed that UISKEI was generally well accepted by the study participants, thereby justifying working towards a new version.

3.2
New version requirements

The main issue in the early version was the paradigm break between stages: while the interface building was done in a pen-based style, the interaction definition was done with lots of mouse clicks on a form. Therefore, the major challenges we wanted to address in the new version were to define interaction in the canvas and to show it to the user in a comprehensible way.

Another improvement was to make the software more customizable, by allowing users to define the collection of elements that could be sketched and recognized. This means that the collection of recognized shapes should be customizable as well and the hard-coded recognition system should be revised.