Enhancing UML Sketch Tools with Digital Pens and Paper

Raimund Dachselt*, Mathias Frisch†, Eike Decker‡
Computational Visualistics / Software Engineering Group
Otto-von-Guericke-University Magdeburg, Germany

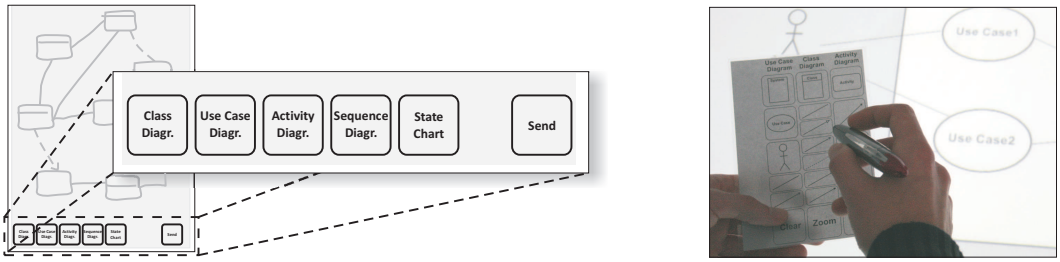

Figure 1: *Left: A page with a sketched diagram. Boxes printed at the bottom can be used to assign UML diagram types to the drawn content. Right: A Paper-palette with UML diagram elements. Elements can be picked up with the pen and placed on a tabletop display.*

Abstract

Drawing diagrams is one of the most important activities during software development processes. A multitude of sketch-based diagram tools were developed to enhance sketching with digital features. However, these applications are not as flexible as paper and often obstruct spontaneous sketching. To overcome these drawbacks we suggest the usage of digital pens and paper in conjunction with sketch-based UML diagram tools. This paper deals with work in progress concerning the seamless integration of paper-based and digital UML-sketching. Amongst others, this includes special UML sketch books and paper palettes used in combination with tabletops.

CR Categories: D.2.2 [Software Engineering]: Design Tools and Techniques—Computer-aided software engineering H.5.2 [Information interfaces and presentation]: User Interfaces—Input devices and strategies;

Keywords: UML, sketching, Anoto, software design

1 Introduction

Especially in early phases of the software design process diagrams are sketched by hand mostly on paper or whiteboards. In the last years many sketch-based diagram tools were developed to support these activities in a digital way (e.g., [Chen et al. 2003]). In general, they try to combine the advantages of pen sketching with the ones of the digital world. So they offer features such as infinite workspaces, zooming of content and the automatic conversion of sketches to formal UML diagrams. Likewise, they aim for a familiar and natural way of interaction with flexible techniques. However, there are still some drawbacks: devices like Tablet PCs or electronic whiteboards are not always available, the devices are not as flexible as paper, need power supply and have to start up which takes time and prevents spontaneous sketching. To overcome these problems, we propose the usage of digital pens and paper in combination with digital sketch-based diagram tools. With this technology, users can work like with common paper, as the pens can be used in single and multi user settings, with the advantages of digital features available at the same time. Moreover, it is possible to realize a seamless transition of sketched diagrams from the paper to interactive surfaces in real time or by subsequent transmission.

*e-mail: dachselt@isg.cs.uni-magdeburg.de
†e-mail: mfrisch@isg.cs.uni-magdeburg.de
‡e-mail: decker@cs.uni-magdeburg.de

2 Related Work

Sketch-based diagram applications such as [Damm et al. 2000], [Hammond and Davis 2002] or [Chen et al. 2003] were developed to support software development processes. They use devices such as Tablet PCs or electronic whiteboards and convert sketches in formal UML diagrams. As stated in the introduction, they still lack the flexibility of common paper. However, there are systems like [Yeh et al. 2007] or [Brandl et al. 2008] which try to integrate paper-based and digital interaction. They support the development of interactive paper applications through authoring and publishing functionalities. In particular, they realize the assignment of services to certain areas of the paper which can be invoked by tapping with the pen and techniques to transfer drawn content to digital devices. Though a multitude of applications were developed with these systems, to our knowledge digital pens and paper were neither applied to diagram drawing in general nor to UML in particular.

3 Digital Pen and Paper Technology

Commonly, digital pens and paper technologies use the Anoto[1] functionality. Equipped with an integrated image processor and an infrared camera, the pens take snapshots of the Anoto dot pattern which is printed almost invisibly on the paper. Every snapshot delivers sufficient information to determine the absolute position of the pen on paper and what the user has written or drawn. Besides that, pens also capture pressure, time and the unique ID of the used paper. There are three ways to transfer data from the pen to an application: by putting the pen in a connected docking station, by tapping with the pen on boxes printed on the page to initialize Blue-

[1]Anoto Group AB, http://www.anoto.com/

tooth transfer or by streaming immediately via Bluetooth. We use this technology not only for paper, but also in conjunction with a rear-projected, multi-pen enabled tabletop display. The Anoto pattern is located underneath its surface and the captured content is streamed to the tabletop. In that way it is possible to draw directly comparable to electronic whiteboards.

4 Combinig UML Sketch Tools with Digital Pen and Paper Technology

4.1 Paper-only approach

Generally speaking, sheets or notepads printed with Anoto-pattern can be used to draw diagrams. The drawn content can be transferred to an application by one of the aforementioned approaches, where the sketches are recognized and converted to formal UML diagrams.

UML sketch books: For that we suggest the usage of special UML sketch books. Besides an area to sketch on, they offer boxes at the bottom of each page representing different diagram types (see Fig.1 (left)). The boxes can be used to assign a certain meaning to the drawn diagram in order to support a proper recognition. By tapping for example on the ”class diagram” box and then tapping the ”send” box, the application tries to convert as many parts of the transferred sketch as possible into class diagram elements, even if they are not drawn in a proper way. It is also conceivable that tapping on one of the boxes is followed by circling certain parts of the sketch. In that way software designers can assign different types of diagrams to the content of a page.

Single sheets of paper: One basic problem with real paper is the lack of space. Digital pen and paper technologies can solve this problem by connecting different sheets of paper. If sufficient space is not available, users can take additional sheets and continue drawing the diagram. Thereby, connections between elements such as relationships of classes can be drawn across different sheets. The system recognizes these connections and the sheets' ID, and thus which diagram parts belong together. This feature is especially beneficial for collaborative settings where several users can connect their diagram parts drawn on different sheets of paper.

4.2 Digital Representation of Diagrams

By putting a sheet of paper on a tabletop displays, content can be dragged from the paper to the display directly. After transferring the drawn content to the tool, the application interprets the content as UML diagrams including handwriting. Parts not recognized stay as sketches and can be corrected manually in the application. As digital pens capture not only what is drawn, but also in which chronological order the content was created, it is possible to visualize how strokes were drawn one by one. This can be beneficial for understanding work processes and to reproduce explanations. Furthermore, digital diagrams can be printed on Anoto-paper, whereupon subsequent changes made by adding (drawing new elements) or deleting (crossing out) elements can be recognized by the system. By transferring the updated diagram back again, the digital version can be synchronized automatically.

4.3 UML-Palettes

We propose the usage of UML-palettes in conjunction with interactive pen-operated tabletop displays. UML-palettes are sheets of Anoto-enabled paper which software designers can hold in their hands. On each palette UML-elements are printed in boxes for example arranged in a grid-layout. By tapping with the pen on one of the boxes, the respective element is ”picked up” and can be placed directly on the tabletop. The diagram elements immediately appear underneath the pen can be moved around and connected with each other by relationships, also selected from the palette (see Fig.1 (right)). In that way it is possible to create ad hoc formal UML diagrams, especially in collaborative work settings. Every user can have his personalized palette and does not need to reach interface elements located far away on the huge interactive display. For people not remembering every UML element in detail, palettes can serve as an additional memory aid. This approach can also be used in combination with sketching. Thereby it is conceivable to adjust elements which are converted in the wrong way or not recognized at all, by picking up the correct element from the palette and placing it on the diagram on the tabletop.

5 Current Prototypes

Currently we implemented two prototypes. One realizes the UML-palette approach and enables multiple users to tap on UML elements on the palette and place them on the tabletop (see Fig.1 (right)). The palettes are laminated sheets with elements arranged in a grid. In addition, it allows switching between interaction modes like zooming, panning and clearing the canvas. We also implemented our own sketch recognition software, based on the Luxinia[2] engine. This prototype can recognize rectangles, ellipses and connections which are drawn directly in the application or which are transferred from a digital pen.

6 Conclusion & Future Work

We presented possibilities to enhance sketch based UML diagram tools with digital pen and paper technologies. Amongst others, we proposed the usage of UML sketch books and UML paper palettes. Our next steps will include the integration of the UML-palette application into the sketch recognition prototype and extending the prototype to assign semantics to UML elements. We also want to test different designs of the UML-palettes and to run user studies, concerning speed, accuracy and user experience.

Acknowledgements This work was partly funded by the ”Stifterverband für die Deutsche Wissenschaft” from funds of the Claussen-Simon-Endowment.

References

BRANDL, P., HALLER, M., OBERNGRUBER, J., AND SCHAFLEITNER, C. 2008. Bridging the gap between real printouts and digital whiteboard. In *In Proc. of AVI 08*, 31–38.

CHEN, Q., GRUNDY, J., AND HOSKING, J. 2003. An e-whiteboard application to support early design-stage sketching of uml diagrams. In *Proc. of HCC ’03*, 219–226.

DAMM, C. H., HANSEN, K. M., AND THOMSEN, M. 2000. Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard. In *Proc. of CHI ’00*, 518–525.

HAMMOND, T., AND DAVIS, R. 2002. Tahuti: A geometrical sketch recognition system for uml class diagrams. *Papers from the AAAI Spring Symposium on Sketch Understanding*, 59–68.

YEH, R. B., KLEMMER, S. R., AND PAEPCKE, A. 2007. Design and evaluation of an event architecture for paper uis: Developers create by copying and combining. In *Stanford University Computer Science Department Technical Report*.

[2] Luxinia Engine, http://www.luxinia.de/