Evalutation of UML CASE Tool with Haptics

Mohamad Eid, Atif Alamri, Jamil Melhem and Abdulmotaleb El Saddik
Multimedia Communications Research Laboratory
University of Ottawa, Canada
{eid, atif, jmelhem, abed} @mcrlab.uottawa.ca

ABSTRACT

Haptics technology has received enormous attention to enhance human computer interaction. Indeed haptics offers a natural user interface based on human gesture system. In this paper, we present a prototype haptic-enabled UML CASE tool that allows software engineering developers to intuitively interact and touch the modeling elements of the tool and feel the force feedback. The tool uses the Omni Phantom device, a common single-point of interaction haptic device. We performed a usability study to compare the haptic-based approach against the traditional mouse-based approach. Even though some users expressed the tiresome of using the haptic device, the study demonstrated the potential of incorporating the haptic modality in UML development tools in terms of user involvement, socialism, and motivation.

Categories and Subject Descriptors

D.3.2 **[Language Classifications]**: Unified Modeling Language

General Terms

Algorithms, Documentation, Design, Standardization, Theory

Keywords

UML, UML CASE TOOLS, haptic, Eclipse, HAML, haptic devices, XML

1. INTRODUCTION

The Unified Modeling Language (UML) is one of the most used visual languages in the software engineering domain for the presentation of object-oriented software design models. UML is supported by CASE tools where the models can be constructed and manipulated as diagrams. Traditional UML CASE tools do not offer enough support for interactivity and motivation [1]. There are many situations where visual feedback is not satisfactory. Some of the issues related to the conventional tools include: visual overloading, limited collaboration and intuitivism, limited interactivity (limited mainly to visual cues), and lack of motivation and physical training. For instance, visual overloading occurs when users are presented with many graphical elements simultaneously. One way to deal with the visual overload is by increasing the number of modalities of feedback cues [2]. This paper investigates the use of haptics feedback as an assistive modeling language in UML CASE tools design.

The science of incorporating force feedback and tactile touch in human computer interaction is referred to as Haptics [3]. In recent years the technology has gained enormous attention in both academia and industry, and several applications are now under investigation. These applications include entertainment and games, arts and media creation, manufacturing, Tele-robotics and Tele-operation, data visualization and education, military simulation and training, and Tele-surgery and rehabilitation. One of the obvious applications of haptics is the enhancement of graphical user interfaces (GUIs).

In addition to the visual feedback, haptics modality can be considered as an extra channel of communication that helps software developers to design UML models in intuitive and entertaining manner. For instance, the resistance and friction provided by stylus-based force feedback adds an intuitive feel of everyday tasks such as dragging, sliding levers, and pressing/depressing buttons. Some of the advantages of incorporating haptics in UML CASE tools include, but not limited to:

- Involvement*:* Haptic modality plays a prominent role in making graphical components physically palpable. Many physical properties such as rigidity, surface texture, softness, viscosity, etc, can be attached to modeling elements. For instance, feeling the weights and collision between modeling elements increases the socialism of the tool.
- Feedback*:* Limitations in screen size and user's ability to cope with a large number of graphic cues have decreased the number of options that are available for the user at the same time. Haptics provides a unique bidirectional and energy-based communication channel between the tool and the developer. Additionally, haptic devices may also help in standardizing the way developers interact with UML case tools by using agreed upon physical stimuli.
- Guidance: A haptic-enabled CASE tool may generate special force feedback stimuli that guide/warn the user whenever s/he performs illegal modeling operation. Furthermore, in learning scenarios, the trainee can be physically guided through the ideal motion by the haptic device.
- Entertainment: The possibility of palpating classes, objects and components is expected to add some sort of excitement to the modeling process.

Our research goal is to enhance the usage of current CASE-tools by incorporating the sense of touch as a new interaction media. The remainder of the paper is organized as follows: section 2 explores related work in the field and highlights the novelty of our proposed tool. Section 3 describes the tool compartments and elements and its software architectural design and implementation. In section 4, we present the evaluation of the tool usability and discuss our findings. Finally, we conclude and summarize the paper in section 5 and recommend possible directions for future development.

2. RELATED WORK

There have been evidences that enhancing the usability of UML CASE tools decreases costs and increases productivity [4]. One

HAS 2008, February 11-14, Quebec, Canada

DOI 10.4108/ICST.AMBISYS2008.2922

way to enrich tool usability is by transferring some of the visual information into audio or haptic outputs [1]. For instance, [5] suggested the use of audition modality to make the UML CASE tool modeling a spoken language. The prototype speech recognition software, named VUML, was developed as a speech interface for UML tools. The authors showed that the incorporation of auditory cues has enhanced the usability of UML CASE tools. To the best of our knowledge, the use of haptic cues in UML CASE tools has not been investigated yet.

Nonetheless, the idea of incorporating haptics modality in GUI-based applications is not brand new. Generally haptic feedback is added to indicate when a user reaches a target, guide the user towards a target, or keep the cursor on a target once reached. Ian Oakle and colleagues [6] developed CHASE, a collaborative haptic-based tool for structured editing and drawing. Three haptic stimuli were implemented: a push effect, a gesture effect, and locate and grab effects. In another example, [7] added haptic effects to GUI features, such as window borders, buttons, and checkboxes. Forces were added to either guide the user towards a target or keep the pointer on a target once reached. Landua and Wells presented in [8] a talky tactile tablet, which combines tactile input with audio information. The proposed interface was claimed to improve the speed and ease of learning.

The incorporation of Haptics in GUI-based applications has shown significant potential as a substitute sensory for visually impaired or blind users. For instance, the Moose prototype system [9] enables blind users to navigate and interact with a desktop application through a GUI window. By moving a haptic mouse over a GUI control object (such as a button, check box, menu, etc.) the user feels differently and eventually can identify the type and current state of the controls (such as the state of a check box). In [10], a user-friendly haptic environment has been developed to allow blind or visually impaired people to access interactive presentations based on HTML Web pages. The application has both tactile and audio feedback.

The architecture and implementation details of the UML CASE Tool were described in [11] with preliminary performance analysis. As a proof of concept, three haptic stimuli were simulated: class gravity, relationship elasticity, and collision detection. Class gravity stimulus enables users to feel the UML class shapes their contents. The relationship elasticity makes the modeling process more realistic and exciting. Finally, collision detection and force feedback enforces the user to modulate his/her design. The usability study suggested several improvements to our design. Consequently, we added new haptic stimuli (namely the element borders and stiffness) to provide more useful and realistic feedback.

3. DESIGN AND IMPLEMENTATION OF THE TOOL

UML CASE tools are complex software systems which incorporate exhaustive features and functionalities. However, we intended to limit our implementation to the portion that will prove the concept and confirm the feasibility of the application. Eventually, we have decided to limit the implementation of the tool to a haptic-enabled class diagram. Briefly, the CASE tool consists of three parts: a drawing area, a palette, and a tool bar (Figure 1). The palette contains icons for the basic modeling elements that the user can manipulate through the haptic device. Basically, two main types of modeling elements are required to build a class diagrams: the class and the reference relation. The tool bar provides the user with basic filling functionalities he/she is allowed to apply to the class diagram. The drawing area contains the user drawn class diagram. The implementation of the drawing area is not easy because it includes the haptic interactions as well.

3.1 Tool Design

A snapshot of the tool interface is shown in Figure 1. Abstractly, the functionalities of the tool are decomposed into two groups: class diagram filing and class diagram manipulation. The former includes basic filing functionality such as new, open, save, and print whereas the latter contains basic requirements for building class diagrams and here is where haptics integration resides. The developer can create classes and relations, delete them, and move them in the drawing area while feeling the various haptic stimuli. There are five haptic stimuli implemented in the tool:

- **Class Weight:** This stimulus enables users to feel the weight of the class depending on the number of its members (class attributes and methods). This allows developers to intuitively manipulate the UML class shapes and feel their contents.
- **Relationship Elasticity:** This stimulus is fired when the user creates a relationship between two classes in the drawing area. The user feels an elastic perception while he/she is creating a UML relation to make the modeling process more realistic and exciting.
- **Class Collisions:** Whenever the tool detects an overlap between existing classes, it sends back a repulsive force feedback that resists the collision action. This makes the user design more sorted and be enforced to modulate his/her design.
- **Component Border:** There are situations where some diagram manipulation functionalities requires locating the edges of an element (such as when locating an element to resize it). Using the mouse, the user has to guess the edge location graphically. Using the border stimulus, the user can easily locate the edge of a component even without any visual feedback.

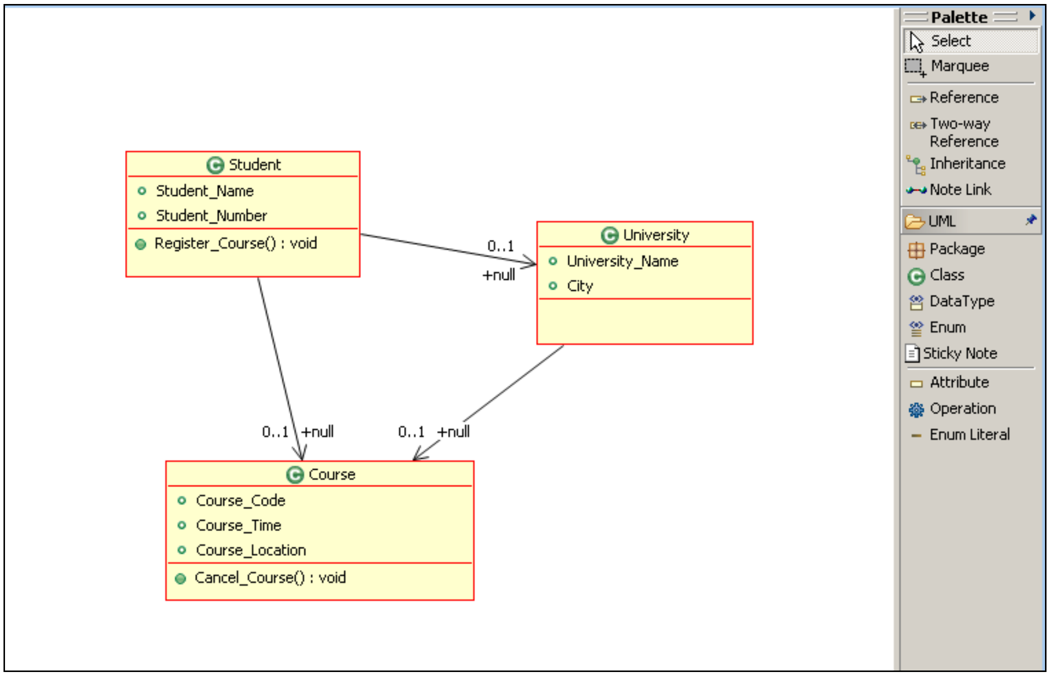

Figure 1: A snapshot of the UML CASE tool GUI.

- **Component Stiffness:** This stimulus is used to differentiate the modeling components so that they are recognized by their stiffness in addition to their geometrical representation. For example, different types of modeling elements can be assigned distinct stiffness so that a developer will find it easier to identify them

3.2 Tool Implementation

The implementation has employed different tools and utilized multiple technologies (see Figure 3). The Rational Software Architect (RSA) has been used for modeling the software architecture of the tool [12]. Eclipse is the platform that was used for coding and building the tool [13]. We used Eclipse because of its extendibility features. Eclipse uses Standard Widget Toolkit (SWT) and JFaces to display GUI elements. SWT is written in Java and presents an alternative to the standard (AWT and Swing) graphical libraries of Java. JFace is a user interface (UI) toolkit with classes for handling many common UI programming tasks. The toolkit simplifies the construction of applications based on SWT and provides actions to define the user's behavior and eventually assigns that behavior to specific components.

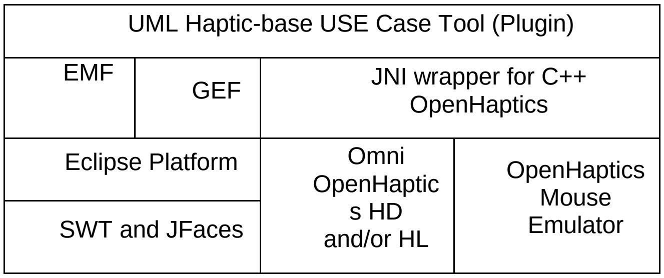

Figure 2: A high level implementation architecture for tool.

The GUI parts of the tool such as drawing area, palette and project explorer are provided by the eclipse platform. The Eclipse Modeling Framework (EMF) is used to provide the UML modeling and diagramming facility whereas the Graphical Editing Framework (GEF) is used to facilitate the display of any graphic model and support interactions from mouse and/or keyboard. All graphical visualization is done via the Draw2D framework, which is a standard 2D drawing framework based on SWT from eclipse.

The haptic device is made accessible for programmers through the OpenHaptics API (HDAPI and HLAPI) [14]. The HDAPI is a low-level foundation layer that is suited for developers familiar with haptic paradigms and rendering. The HLAPI is built on top of the HDAPI and supports high level scene rendering. It is desired for quick and easy haptic applications development. We used the JNI wrapper to wrap the OpenHaptics API and invoke native API methods.

4. PERFORMANCE EVALUATION

The objective of the evaluation is to measure the quality of the designed haptic-enabled UML tool and compared it to the mouse-based tools.

4.1 Experiment Setup

Ten subjects, with various experience backgrounds in haptic devices, participated in the experiment. The experiment comprises drawing two basic UML diagrams, namely an employee-project diagram and a student-course diagram. The first exercise was performed initially using the mouse and then repeated using the haptic device. As per the second exercise, the subject was required to draw the diagram in the reverse order; first with the haptic device and subsequently with the mouse. The reason why was to evaluate the perceptual effects of switching from haptic to mouse approaches or vise versa. Finally, to eliminate any biasing, the second task was performed two weeks after the first one.

The experiment session was divided into three parts: a preparation session, a practicing session, and a testing session. The preparation session included an illustration of the purpose and goals of the test along with a short demonstration of the tool. The practicing session included familiarization with the haptic device and various haptic stimuli by performing simple drawing exercises. The testing session was where the subject was asked to draw a class diagram with and without haptic feedback (using the haptic device and the mouse). The test subjects were asked to practice and utilize the five haptic stimuli specified earlier. Users' experience is measured and compared by exploiting a questionnaire that reflects the personal opinion on the tool.

4.2 Tasks

The two modeling exercises were designed to measure the feasibility of using haptic feedback in UML CASE tools. As shown in Figure 3, the employee-project diagram comprises two classes with a simple reference relationship. The two classes had significant differences in their contents so that the user can clearly notice the difference in their gravity and collision. This helps subjects to better evaluate the force feedback provided by the tool.

In the student-course modeling exercise, the focus was to increase a bit the complexity of the diagram and measure the values of other stimuli such as relationship elasticity and element's borders. Furthermore, the complexity of the diagram has a bit increased to reflect more realistic class diagrams.

Figure 3: The employee-project modeling exercise

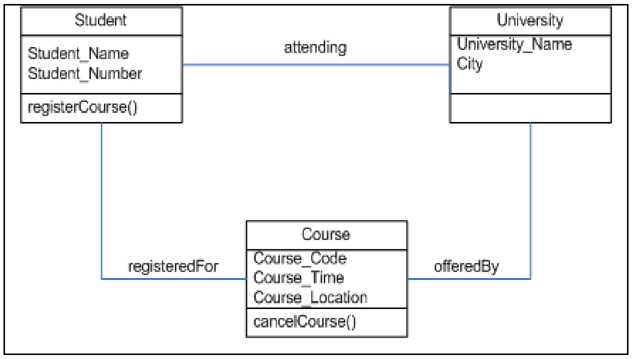

Figure 4: The student-course modeling exercise.

4.3 Results

Measuring the usability of the haptic-based UML CASE tool is not a straightforward task. Restricted by the preliminary stage of this work, we chose to use a performance metric that includes one quantitative measure, namely the task completion time (TCT) and two qualitative measures (the ease of usage and the user involvement). The two subjective measures were estimated by analyzing the user responses to the questionnaire.

As per the average TCT per subject, we found that, for both modeling exercises, the users took longer time to complete the tasks using the haptic device than using the mouse (see Table 1). For example, the average TCT of the student-course model was 6.5 minutes with haptic feedback and 3.5 minutes using traditional mouse. This significant difference was expected, as users are already very well trained and comfortable with using a mouse, while a haptic device is much newer and therefore not as intuitive. Subsequently, after the users had gained some experience in using the haptic device, the difference in TCT with and without haptic feedback was greatly decreased.

Table 1. TCT (minutes) with and without force feedback

	Student-Course		**Employee-project**	
	With haptics	Without haptics	With haptics	Without haptics
Subject 1	5	3	7	6
Subject 2	6	3	4	4
Subject 3	4	4	5	4
Subject 4	9	5	7	5
Subject 5	8	3	5	4
Subject 6	8	4	6	4
Subject 7	9	3	4	4
Subject 8	5	3	5	4
Subject 9	7	4	5	3
Subject 10	4	3	5	3
Average	**6.5**	**3.5**	**5.3**	**4.1**

In order to measure the user involvement and socialism of the tool, we analyzed the users' feedback and found that 70% of the participants felt that the tool has increased the interactivity of the tool. Additionally, 76% of the users agreed that haptic feedback has increased their experience and the overall quality of the tool. On the other hand, 70% of the users said that using the haptic device has caused hand fatigue. The reason was due to the device back-drivability and arm tiredness. One of the suggestions was to use an arm support to compensate for the device inertia and unmask the haptic stimuli.

5. CONCLSION AND FUTURE WORK

The current paper presented the development and evaluation of a haptic-enabled UML CASE tool. We compare the haptic-based CASE tool usability versus that of a mouse tool. It is found that the incorporation of haptics modality has enhanced the tool usability and increased its interactivity. We anticipate that the widespread and availability of haptic devices to the public will play a critical role in making such applications a successful experience. The introduction of the low-cost Falcon device [15], by Novint Technologies is a step further towards this goal.

As per future work, we are planning to build a shared collaborative UML CASE tool. This will considerably increase the socialism of the UML development process. Furthermore, the use of other multiple-point of interaction haptic devices enables more realistic interaction with the tool. For example, the CyberForce station haptic device allows users to grab a modeling object and move it around in a more intuitive way. Finally, we would like to define different categories of force feedback depending on the type of diagram to be developed while keeping the consistency between the haptic stimulus and its semantic meaning.

6. ACKNOLDGMENT

This research is partially supported by NSERC, ORNEC, and IBM Ottawa Software Lab.

7. REFERENCES

[1] K. Lyon and Peter J. Nürnberg, Multimodal Modelling Languages to Reduce Visual Overload in UML Diagrams. *Proceedings of the Third Program Visualization Workshop*, Research Report CS-RR-407, Department of Computer Science, University of Warwick, UK, 2004.

[2] G. Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich, “Data mountain: using spatial memory for document management” In Proceedings of the 11th annual ACM symposium on User interface software and technology, pages 153-162, 1998.

[3] M. Eid, M. Orozco and A. El Saddik, “A Guided Tour in Haptic Audio Visual Environment and Applications”, International Journal of Advanced Media and Communication, vol.1, n.3, 2007.

[4] B. Myers “Challenges of HCI Design and Implementation” in Interactions, pp 73-83, 1994.

[5] S. Lahtinen, and J. Peltonen , “Enhancing usability of UML CASE-tools with speech recognition”, In Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments (October 28 - 31, 2003). HCC. IEEE Computer Society, Washington, DC, 227-235.

[6] I. Oakley, S. Brewster P.D. Gray, "Can You Feel the Force? An Investigation of Haptic Collaboration in Shared Editors" EuroHaptics'2001, Birmingham, UK, 2001.

[7] T. Miller and R. Zeleznik. “An insidious haptic invasion: Adding force feedback to the X desktop” In Proceedings of ACM UIST 1998, pages 59–64, 1998.

[8] S. Landua, and L. Wells, “Merging Tactile Sensory Input and Audio Data by Means of The Talking Tactile Tablet” in Proc. Eurohaptics 2003. Dublin, Ireland, 2003.

[9] S. O'Modhrain and R. B. Gillespie. “The Moose: A Haptic User Interface for Blind Persons”, In Proceedings of the Third WWW6 Conference, Santa Clara, CA, April 1997.

[10] G. Nikolakis, I. Tsampoulatidis, D. Tzovaras, and M.G. Strintzis, "Haptic browser: a haptic environment to access HTML pages", In SPECOM-2004, 514-520, 2004.

[11] A. Alamri, M. Eid, and A. El Saddik. “A Haptic Enabled UML CASE Tool” in Proc. Of IEEE ICME 2007.

[12] Rational Software Architect RSA. http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html

[13] Eclipse http://www.eclipse.org

[14] http://www.sensable.com/products-openhaptics-toolkit.htm.

[15] Falcon device, Novint Technologies: http://www.novint.com.