Possible User Interfaces for Future Human Robot Interaction

HRI workshop

Eliezer Kantorowitz
CS Department, Technion
Haifa 3200003
Israel
kantor@cs.technion.ac.il

Introduction

The user interfaces (UI) employed in HRI may combine such modalities as verbal, graphical, and haptic. An interesting HRI development is based on the observation by Jessica R. Cauchard et al that "people interact with drones as with a person or a pet" [2]. It was shown that "people felt comfortable instructing a drone with gestures for such activities as precise landing, for coming closer to the person and for stopping" [3]. In other experiments "the drones where given a combination of personality (adventurer, anti-social or exhausted) and an emotional state (Dopey, Sad, Sleepy, Grumpy, Shy or happy). The personality and emotion were manifested by their motions. For example the adventurer flew higher than the exhausted. The human experiment participants observed the drones doing different tasks and had to recognize the personality, mood and task. The human participants were able to "identify the behavior of the drone by observing its physical movement and its response to commands. Participants managed to accurately associate this behavior to an emotional state corresponding to a personality model. We believe that drones are a viable platform to become accepted sociable entities ... the recognized emotions might be used to inform users of the drone's intentions and convey meaningful feedback that would be hard to convey otherwise" [6]. "It thus possible that some future HRI will be a combination of reading the emotions of the robot, verbal, graphical, voice and haptic messages. This is similar to communication between human which includes reading the body language". The possible future UI for HRI, as expected by Cauchard et al, are more complex and difficult to design than many common state of the art UIs. In order to facilitate the difficult design of such UIs it is suggested to employ the known approach of first focus on what to do (requirements elicitation) and later to focus on how to it. This approach was implemented in our semantic user interface [7] [8], which is the subject of this paper.

Let me first explain how we arrived to the idea of employing semantic user interfaces. Many years ago my student Yoram Hazan compared two different graphical user interfaces (GUIs) [5]. One GUI was easy-to-understand and the other was difficult-to–understand. Those who used the difficult-to-understand GUI got angry or frustrated in some experimental situations. Those who used the easy-to-understand GUI asked sometimes after completing a task of the experiments, whether they may try to do the task in a different way. These user requests suggest both that they have achieved a high level of familiarity with the application and that they are comfortable with it. There is also the possibility that the application will be exploited by some users in a creative and especially productive way. One of my conclusions was that a semantic UI that only specifies the interactions and not the technicalities of its implementations may facilitate the understanding of the applications. Therefore, I suggested to my student A. Lyakas to investigate the feasibility of semantic user interfaces [**Error! Reference source not found.** [8]. In this paper we do not employ our original formulation of the method, but a formulation based on the terminology of Chameleon Reference Framework (CRF) [1], which may become a W3C recommendation [9]. In this description we employ HRI UI examples.

Abstract Semantic UI

An abstract *semantic UI* is composed of a number of abstract *semantic control*s, each of which specifies a human computer interaction. This specification of a semantic control specifies only what the interaction does, for example "a user instructs a moving robot to stop". The specification of a semantic control does, however, not specify how the interaction is *implemented.* An interaction may be implemented in a number of different ways. Each one of these implementations is done by a separate *concrete semantic control.* A concrete control differs from an abstract control by being executable. Consider for example the abstract semantic control:

"A user instructs a moving robot to stop and it stops"

It may be implemented by the concrete semantic control:

"The user employs a stop gesture which is interpreted by the robot and it stops"

An alternative concrete semantic control:

"The user employs a smartphone application to order the robot to stop. The robot gets the order wirelessly and stops".

There may be further implementations of this abstract semantic control.

Figure 1 illustrates the relationship between the semantic control and its implementations:

Abstract Semantic Control	"*Stop the robot"*
Possible implementations by different concrete semantic controls	"Stop robot by gesture"
	"Stop robot by application"
	Possible further implementations

Figure 1. The abstract semantic control "Stop the robot" and two of its concrete implementations. Note that the abstract semantic control "Stop the robot" does not specify how to implement the interaction, while the two concrete control specifies the "how" (by gesture and by application).

An *abstract semantic control is said to be an abstraction of all its possible implementations.*

In the first step of the two step UI design method [10] the code of the application is only provided with an abstract semantic UI, which is not executable. The abstract semantic UI provides, however, a specification of the interactions that should be implemented by the concrete controls of the concrete UI. This specification is employed at the second step of the process by the designer of the concrete UI. This UI designer selects appropriate implementations for all the abstract semantic controls and add required facilities such as means for navigating in the UI.

DISCUSSION AND FUTURE RESEARCH

Our semantic interface is defined, such that any feasible kind of interaction can be specified by a semantic control. This design is motivated by the complex multi modal kinds of interactions that may be expected in future robots. The semantic UI may be useful in experimental research. Consider for example an experimental comparison of the user experience with two different implementations of a particular semantic control, for example compare stopping the robot using gesture versus using a smartphone application.

It may be interesting to investigate whether the MDE approach may be employed for adapting a semantic UI to a particular use case.

An interesting result of our research is the introduction of the abstract semantic UI that only specifies the possible interactions, but not their implementations. This abstraction may be useful in studies where the methods of implementation of the interactions play a minor role.

The components of the widely employed GUIs are such geometrical entities as buttons and menus. An abstract semantic interface is composed of number of abstract semantic controls. A behavioral study of how the UI designers and the UI users manage these two different UI types is planned.

REFERENCES

[1] G. Calvary. J. Coutaz. D. Thevenin. Q. Limbourg. L.Bouillon and J. anderdonckt. 2003. A unifying reference framework for multi-target user interfaces. Journal of Interacting With Computers. 15:3. 289-30

[2] Jessica R. Cauchard, Jane L. E, Kevin Y. Zhai, James A. Landay. 2015. Drone & Me: An Exploration into Natural Human-Drone Interaction. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15), 361-365

[3] Jessica R. Cauchard, Kevin Y. Zhai, Marco Spadafora, James A. Landay. 2016. Emotion Encoding in Human-Drone Interaction. 263-270

[4] Coutaz. User Interface Plasticity: 2010, Model driven Engineering to the limit. Proceedings of the 2nd ACM SIGCHI symposium on engineering interactive computing systems. 1-8
[5] Yoram Hazan. 1989. A Comparison between the Facilitative Effects of a Domain-Oriented Presentation User Interface and a General-Purpose Presentation User Interface on Learning Computer Applications, PhD thesis, Faculty of Science and Education, Technion-Israel Institute of Technology, Haifa, Israel.
[6] E, Ilene, L. E. James, A. Landay, Jessica R. Cauchard. 2016. Drone & WO: Cultural Influences on Human-Drone Interaction Techniques. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). 6794-6799
[7] E. Kantorowitz and A. Lyakas. 2005. Use-Case Components for Interactive Information Systems. Science of Computer programming. 56, 5-21
[8] E. Kantorowitz. A. Lyakas (2012). Environment Independent Software. U.S. Patent 8.332.830
[9] Gerrit Meixner, Gaëlle Calvary, Joëlle Coutaz, Introduction to Model-Based User Interfaces, January 2014, W3C, https://www.w3.org/TR/mbui-intro/.Retrieved August 2017
[10] E. Kantorowitz. 2016. Semantic User Interface Controls.. European Conference on Cognitive Ergonomic (ECCE 2017). ACM Digital Library