M. Koch, A. Butz & J. Schlichter (Hrsg.): Mensch und Computer 2014 Workshopband,
München: Oldenbourg Wissenschaftsverlag, 2014, S. 3-6.

A Framework for Controlling Robots via Brain-Computer Interfaces

Ronny Seiger, Tobias Nicolai, Thomas Schlegel

Software Engineering of Ubiquitous Systems, Technische Universität Dresden

Abstract

The application presented in this paper employs a brain-computer interface for controlling a service robot in an ambient assisted living setting. The application is based on a flexible software framework that allows for the on-the-fly definition of event-to-action mappings. It provides means for integrating interaction events from multiple sources and thereby facilitates the rapid development and test of interaction concepts for multi-modal interactions.

1 Introduction

Modern interaction devices gather and process information from multiple sensors to provide new means and modalities for users to interact with digital systems. However, the interaction concepts describing a mapping of user interactions to actions that will be triggered within an application are rather static. This requires the user to be able to perform and to learn these predefined interactions. To address this issue, manufacturers of interaction devices started providing software development kits (SDKs) that enable developers to create custom applications. This leads to more flexible interactive systems that can be suited to the end-users' needs. However, the SDKs are usually proprietary solutions tied to one interaction device.

Brain-computer interfaces (BCIs) have a great potential to support people with physical disabilities within ambient assisted living scenarios. Nevertheless, people have different abilities and requirements regarding the interaction with their physical and virtual surroundings. This usually results in a high effort to create highly customised applications and environments. Changing and adjusting the software for controlling these environments often requires the applications to be shut down and modified according to the altered conditions.

The BCI used in this work provides means for interaction via thoughts, facial expressions, emotions and a gyroscope. Applying these modes of interaction, the user is able to control the basic movement of a robot. We chose this scenario as it clearly shows the need for a framework enabling the customisable definition of an interaction concept with respect to the user's abilities and preferences. Not all users will be able to use the entire set of available interactions and the quality of interaction events recognised by the BCI will vary among users. Therefore, the mapping between events and actions to be executed by the robot has to be defined per user. The adjustment of these user profiles should require little effort and be possible at runtime.

2 Framework Architecture

When developing the software architecture for the interaction framework, we tried to adhere to a simple class structure for managing and connecting sources of interaction events with functions provided by physical and virtual devices. The class structure can be found in Fig. 1.

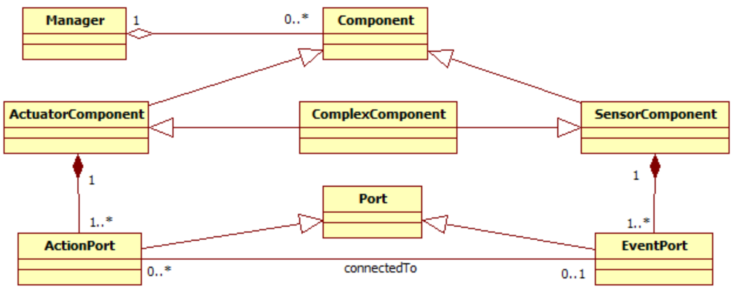

Figure 1: Framework class structure

The central concept of the system is the *Component*, which can be a physical device but also a web service or a desktop application. A component is either a *SensorComponent* producing interaction events or an *ActuatorComponent* consuming interaction events and executing an action. *ComplexComponent*s can play both roles. A sensor component contains a list of *EventPort*s, each of which representing an interaction event that can be triggered by the sensor component. Similarly, an actuator component maintains a list of *ActionPort*s corresponding to a particular function that the component is able to execute. The activation of an event port will lead to the activation of the action ports connected with this port and trigger the execution of the corresponding actions. Multiple event ports and multiple action ports can be active at the same time. In that way, multi-modal interactions by one or more interaction devices are enabled. Our prototype currently supports binary port states (active/inactive).

The concept of a *ComplexComponent* allows for the integration of more complex devices providing both sensor and actuator functionalities. By introducing specialized components as extensions of complex components, we are able to deduce higher level events from one or several sensor components. In that manner logical and temporal dependencies between multiple events triggering a specific action can be implemented.

Currently, the types of sensor components and their event ports have to be integrated at design time. The functionality provided by the actuator components can be modified at runtime as the prototype supports web service calls. Upon importing a WSDL compatible description of the actuator component's service, an action port is generated automatically for each method.

3 Implementation

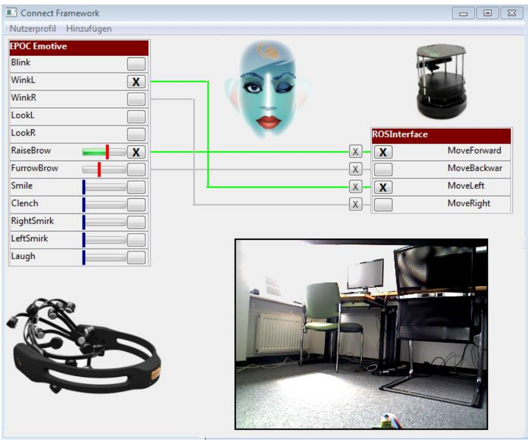

Figure 2: Compact user interface of the demonstrator application

An **Emotiv EPOC** BCI represents the *SensorComponent* within our system. Its SDK provides access to high level events regarding thoughts, emotions, facial expressions (*EventPorts*) already interpreted and also to the build-in gyroscope sensor. A **TurtleBot 2** robot represents the *ActuatorComponent* that will be controlled. Its movement functions have been abstracted to enable the robot to drive forwards, backwards, to the left and to the right (*ActionPorts*).

The demonstrator application provides a **user interface** for managing sensor, actuator and complex components. Upon loading the profiles for the components, the ports of the sensor and actuator components can be connected graphically. These mappings can be created and changed at runtime and saved in individual profiles. The user receives visual feedback about the state of the events recognized by the sensor component and about the actions performed by the actuator components (see Fig. 2).

4 Related Work

Kübler et al. argue that applications employing BCIs have to be designed from a user-centric perspective taking into account the user's needs (Kübler et al. 2013). In comparison with widely-used icon and symbol spelling approaches, we are aiming at providing a more direct way of interaction via customisable interaction concepts and thereby focussing on the user's preferences. In (Hoste et al. 2011) the authors describe an advanced framework for fusing multi-modal interaction events on different semantical levels using a formal declaration language and inference mechanisms. As we will use our interaction framework to deploy applications in smart home environments, our goal was to design a simple framework that can be extended and configured easily at runtime and still integrate multiple devices in a user-friendly way. In (Wijayasekara & Manic 2013) a simple robot control application based on an Emotiv EPOC BCI is presented and evaluated. The results show that the control solely with the help of the BCI's thought recognition is very imprecise. Therefore, our framework allows for the combination of multiple interaction devices and modalities.

5 Discussion & Future Work

The framework presented in this demo paper enables the flexible and easy definition and test of interaction concepts at runtime. The BCI provides a large set of possible interactions that can be used for controlling an application. However, the carrier of the BCI may prefer some interactions over others. Our framework supports the easy customisation and individual storage of user preferences. These settings can be changed and modified at runtime.

Although the interaction devices (sensor components) can be easily extended via inheritance, we are planning to support a more flexible way of integrating new interaction events and interaction channels at runtime (Keller et al. 2013). The set of available components (sensor, actuator, complex) will be extended to enable more complex scenarios of intelligent environments controlled by multi-modal interactions.

References

Kübler, A., Holz, E., Kaufmann, T. & Zickler, C. (2013). A User Centred Approach for Bringing BCI Controlled Applications to End-Users. In: *Brain-Computer Interface Systems - Recent Progress and Future Prospects*.

Hoste L., Dumas B. & Signer, B. (2011). Mudra: a unified multimodal interaction framework. In *Proceedings of the 13th international conference on multimodal interfaces (ICMI '11)*. ACM, New York, NY, USA, pp. 97-104.

Wijayasekara, D. & Manic, M. (2013), Human machine interaction via brain activity monitoring. In *Human System Interaction (HSI), 2013 The 6th International Conference on* , IEEE, pp.103-109.

Keller, C., Kühn, R., Engelbrecht, A., Korzetz, M. & Schlegel, T. (2013). A Prototyping and Evaluation Framework for Interactive Ubiquitous Systems. In Streitz, N. & Stephanidis, C. (Eds.): *Distributed, Ambient, and Pervasive Interactions, LNCS*. Berlin Heidelberg: Springer, pp. 215-224.