A MULTI-MODAL HUMAN-MACHINE-INTERFACE FOR ACCELERATOR OPERATION AND MAINTENANCE APPLICATIONS

R. Bacher, DESY, Hamburg, Germany

Abstract

The advent of advanced mobile, gaming and augmented reality devices provides users with novel interaction modalities. Today's accelerator control applications do not provide features such as speech, finger- and hand-gesture recognition or even gaze detection. Their look-and-feel and handling are typically optimized for mouse-based interactions and are not well suited for the specific requirements of more complex interaction modalities. This paper describes the conceptual design and implementation of an intuitive single-user, multi-modal human-machine interface for accelerator operation and maintenance applications. The interface seamlessly combines standard actions (mouse), actions associated with 2D single- / multi-finger gestures (touch sensitive display) and 3D single- / multi-finger and hand gestures (motion controller), and spoken commands (speech recognition system). It will become an integral part of the web-based, platform-neutral Web2cToGo framework belonging to the Web2cToolkit suite and will be applicable for desktop and notebook computers, tablet computers and smartphones, and even see-through augmented reality glasses.

INTRODUCTION

In a control room the mouse is the standard user input device to interact with accelerator control applications. Being well accepted by the operators it provides a very accurate pointing capability even onto complex applications with a wealth of graphical widgets. However, hardware commissioning and maintenance use cases might profit from novel interaction capabilities (modalities). For instance the alignment of mirrors mounted on an optical table to adjust a laser beam spot often requires a "third hand". Interacting with control applications via spoken commands could be an appropriate alternative. Likewise remote-controlling a head-mounted display showing some documentation or control application panels might be considerably simplified by recognizing spatial gestures such as clenching a fist or snapping fingers.

This paper reports the conceptual design and the implementation of a single-user human-machine-interface (HMI) for accelerator operation and maintenance applications in the context of the Web2cToolkit Web service collection [1,6]. Web2cHMI defines a set of common user interactions associated with various modalities including flat or spatial gestures, spoken commands, and ordinary mouse or touch actions. This approach applies to desktop or notebook computers with passive or touch-sensitive display, tablet computers or smartphones, and even see-through Augmented Reality glasses.

Web2cToolkit is a collection of Web services including among others Web2cViewer and Web2cToGo. Both Web applications implement Web2cHMI. The Web2cViewer Web service provides a user-configurable interactive synoptic live display to visualize and control accelerator or beam line equipment. The Web2cToGo client application is especially designed for mobile devices and is capable of running simultaneously up to 15 multi-page Web2cToolkit-compliant Web applications such as Web2cViewer applications. At any particular time the selected page of the selected user application is displayed while other pages of the same application and the pages of the other applications are hidden. The Web2cToGo client application provides three different views including Explorer View, Navigation View, and Operation View.

CONCEPTUAL DESIGN AND BASIC FEATURES

Web2cHMI is a set of JavaScript (JS) classes to be used in Web client applications executed by a Web browser. A Web2cHMI-compliant browser must implement the Web Sockets API and the getUserMedia / Stream API (required for speech recognition only). Table 1 summarizes the implementation status [2] by major Web browsers (most recent desktop or mobile version).

The recognized user input from each supported modality is handled by its own modality-specific class notifying a common interface class which translates the modality-specific user actions into common unique application-specific commands for further execution.

Table 1: Implementation Status of Required JS API

Web Browser	Web Sockets API	getUserMedia / Stream API
Edge	yes	yes
Firefox	yes	yes
Chrome	yes	yes
Safari	yes	no
Opera	no	yes
iOS Safari	yes	no
Opera Mini	no	no
Android Browser	yes	yes
Chrome for Android	yes	yes

ISBN 978-3-95450-148-9

Supported Modalities

Web2cHMI supports five different modalities which can be used simultaneously:

- 1D/2D flat gestures including single-finger actions (**mouse**) and single- or multi-finger gestures (**touch-sensitive display**)
- 2D/3D spatial gestures including hand-gestures (**LEAP Motion controller** [3]) and hand- or arm-gestures (**Myo gesture control armband** [4])
- English Spoken commands (**Sphinx speech recognition** [5]).

Table 2 summarizes the support status of the various modalities on major operating systems (most recent version).

Table 2: Support Status of Web2cHMI Modalities

OS	Mouse	Touch	LEAP	Myo	Speech
Win	yes	yes	yes	yes	yes
Linux	yes	yes	yes	no	yes
MacOS	yes	yes	yes	yes	yes
Android	no	yes	no	yes	yes
iOS	no	yes	no	yes	yes

Web2cHMI recognizes various primitive, i.e. native or input device-specific gestures including

- Mouse: Click, Move
- Touch-sensitive display: Tap, Move / Swipe, Pinch (two fingers)
- LEAP Motion controller: Key-Tap, Swipe, Open-Hand, Closed-Hand, Circle
- Myo gesture control armband: Double-Tap, Wave-Out / Wave-In, Fingers-Spread, Fist

In addition, enriched gestures, i.e. primitive gestures followed by moves, rotations etc. are supported. If applicable or required by ergonomics principles, different gestures may be applied by right or left handed individuals.

Spatial Gestures

The LEAP Motion controller (Figure 1) and the Myo gesture control armband (Figure 2) are connected locally through USB and Bluetooth, respectively. The raw sensor signals are processed by a local Web server which communicates with the Web2cHMI via Web Sockets.

Unlike mice or touch-sensitive displays gesture recognition devices such as LEAP or Myo do not allow an accurate positioning of a cursor.

The LEAP Motion controller can also be used in upside-down orientation.

To avoid unwanted responses to unintentional gestures recognition ability must be armed or disarmed explicitly by the user. Arming and disarming the LEAP Motion controller is performed by the Key-Tap gesture. While armed the application displays a moving label (yellow = finger is not in the sensor's active range, green = finger is in the sensor's active range) indicating the position of the pointing finger within a virtual frame (width = 200 mm, height = 200 mm) located 100 mm above the sensor. The active sensor range covers ± 100 mm before and behind the sensor, respectively.

Figure 1: LEAP Motion sensor

In order to arm and disarm the Myo gesture control armband the user has to execute a Double-Tap gesture. While armed the application displays a moving green label indicating the position of the hand of the arm wearing the sensor within a virtual frame (width = 500 mm, height = 500 mm). During arming the arm direction is taken as the zero-reference and the label position is set arbitrarily to the center of the application window.

If a gesture is successfully recognized the next gesture recognition is inhibited for approximately 1s while the label is fixed to the center of the application window.

Figure 2: Myo gesture control armband

Spoken Commands

Phonetic analysis of locally recorded audio sequences is performed by the Web2cToGo server application [6] which notifies the Web2cHMI via Web Socket communication if a spoken command has been recognized. In order to preserve privacy audio recording can be paused or resumed by clicking the speech recognition icon at the left border of the Web2cToGo application window. To avoid unwanted responses to unintentional spoken commands the user must say "Ok" or "Sleep" to arm or disarm the ability to recognize spoken commands, respectively.

ISBN 978-3-95450-148-9

Figure 3: Web2cToGo Explorer View

EXPLORER VIEW HMI

The Web2cToGo Explorer View (Figure 3) displays a set of icons to select a user application. It provides launching or displaying of already launched but currently hidden user applications. In addition it indicates the currently available modalities and the corresponding arming status (L = LEAP, M = Myo and S = speech recognition).

Amongst other user input Web2cHMI recognizes for instance the following corresponding actions to launch or display the selected application:

- Mouse: **Click** selected application icon
- Touch-sensitive display: **Tap** selected application icon (right or left hand)
- LEAP Motion Controller: **Open-Hand** (right or left hand)
- Myo gesture control armband: **Fingers-Spread** (right or left arm)
- Speech Recognition: **Say** "Display"

NAVIGATION VIEW HMI

The Web2cToGo Navigation View (Figure 4) previews the selected user application. It allows switching between user applications, switching to Explorer or Operation View, closing the current user application, browsing between application pages, and zooming, resizing or scrolling pages of user applications.

Among other user input Web2cHMI recognizes for instance the following corresponding actions to zoom in the current application page:

- Mouse: **Click** magnifier icon
- Touch-sensitive display: **Tap** magnifier icon (right or left hand) or Right/Left-Tilted Open-**Pinch** (right/left hand)
- LEAP Motion Controller: **Open-Hand** & Clockwise Rotation (right or left hand)
- Myo gesture control armband: **Fingers-Spread** & Clockwise rotation (right or left arm)
- Speech Recognition: **Say** "Zoom In"

Figure 4: Web2cToGo Navigation View

ISBN 978-3-95450-148-9

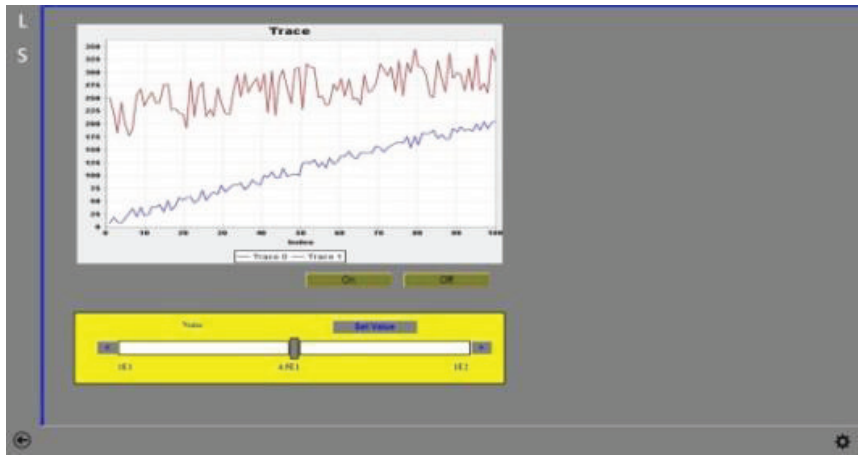

Figure 5: Web2cViewer Operation View

OPERATION VIEW HMI

While switched to Operation View (Figure 5) the Web2cToGo client application displays a Web2cToolkit-compliant Web application. The Web2cViewer application implements Web2cHMI. Each Web2cViewer application page might contain one widget instance of each of the following interactive widget types, only. According to its type, an interactive widget is capable of performing a specific, predefined user action such as opening a vacuum valve or changing a set value of a power supply:

- On-type Button (user action = "On")
- Off-type Button (user action = "Off")
- Slider (user action = "Set Value")
- Chart (user action = "Zoom Data")

In addition a page might contain an unlimited number of passive Web2cViewer widgets such as labels or value fields.

Among other user input Web2cHMI recognizes for instance the following corresponding actions to increase a set value of an attached controls device in small steps using the slider widget:

- Mouse: **Click** ">"-button and **Click** "Set Value"-button of the slider widget
- Touch-sensitive display: **Tap** ">"-button and **Tap** "Set Value"-button of the slider widget (right or left hand)
- LEAP Motion Controller: Clockwise **Circle** (right or left hand)
- Myo gesture control armband: **Fist** & Clockwise Rotation (right or left arm)
- Speech Recognition: **Say** "More"

PROJECT STATUS AND OUTLOOK

While being fully implemented by Web2cToGo and Web2cViewer Web2cHMI is still experimental. An implementation by Web2cArchiveViewer is envisaged. In general the performance and reliability of gesture and speech recognition has to be improved. The defined gestures or spoken commands will likely be changed in future based on usability experiences gained in a real accelerator environment.

REFERENCES

[1] Web2cToolkit; http://web2ctoolkit.desy.de
[2] http://www.caniuse.com
[3] LEAP; https://www.LEAPmotion.com
[4] Myo; https://www.myo.com
[5] Sphinx-4; http://cmusphinx.sourceforge.net/sphinx
[6] R. Bacher, "Renovating and Upgrading the Web2cToolkit Suite: A Status Report", PCaPAC'14, Karlsruhe, Germany, October 2014, FCO2012, p. 234 (2014); http://www.JACoW.org

ISBN 978-3-95450-148-9