# ENHANCING THE MAN-MACHINE-INTERFACE OF ACCELERATOR CONTROL APPLICATIONS WITH MODERN CONSUMER MARKET TECHNOLOGIES

R. Bacher, DESY, Hamburg, Germany

*Abstract*

The paradigms of human interaction with modern consumer market devices such as tablets, smartphones or video game consoles are currently undergoing rapid and serious changes. Device control by multi-finger touch gesture or voice recognition has now become standard. Even further advanced technologies such as 3D-gesture recognition are becoming routine. Smart enhancements of head-mounted display technologies are beginning to appear on the consumer market. In addition, the look-and-feel of mobile apps and classical desktop applications are becoming remarkably similar to one another. We have used Web2cToGo to investigate the consequences of the above-mentioned technologies and paradigms with respect to accelerator control applications. Web2cToGo is a framework which is being developed at DESY. It provides a common, platform-independent Web application capable of running on widely-used mobile as well as common desktop platforms. This paper reports the basic concept of the project and presents the results achieved so far and discusses the next development steps.

## INTRODUCTION

Today's consumer market turns out to be a major technology driver. In particular smartphones and tablets, game consoles (e.g. Microsoft Xbox / Kinect [1]) or augmented reality devices (e.g. Google Glass [2]) have popularized novel features in a wide community of users. The "App" is now established as an application class. Staying online everywhere all the time is a common practice and essential for using popular services such as social network portals. In addition, versatile and powerful man-machine interfaces (MMI) providing touch gesture (2D) and motion gesture (3D), as well as speech and gaze recognition are available. This paper deals with these interface techniques and their potential use in accelerator operations and maintenance cases.

Control room operators or service technicians might benefit from advanced options to interact with computers. They might

- select, open and navigate between application windows or change the display sizes by using single- and multi-touch gestures or spoken commands,
- bring application windows to foreground and keep control merely by gazing at an application window,
- scroll through synoptic views by turning their heads or moving their eyes,
- take a snapshot of an operations screen in the control room or even of the current accelerator status by performing a grabbing gesture,
- control the content of over-head displays by using motion gestures or spoken commands,
- select or interact with accelerator devices without hand contact via spoken commands, gazing onto device-specific widgets or pointing towards a widget projected onto the palm of their hands or onto a wall in front of them,
- trigger or interfere with controls procedures by snapping their fingers e.g. in order to inject beam or by thumbing up, down, left, or right e.g. in order to decrease, increase, accept or discard a set value.

It is even likely that the next generation of operators will no longer be familiar with detailed, mouse-controlled applications.

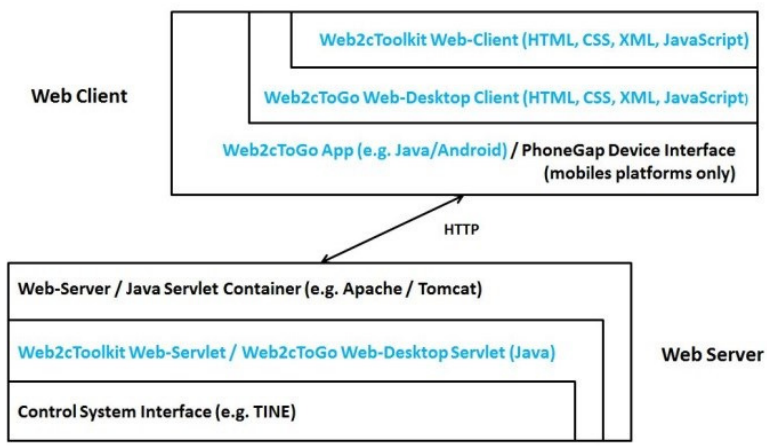


Figure 1: Web2cToGo client-server architecture.



ISBN 978-3-95450-139-7

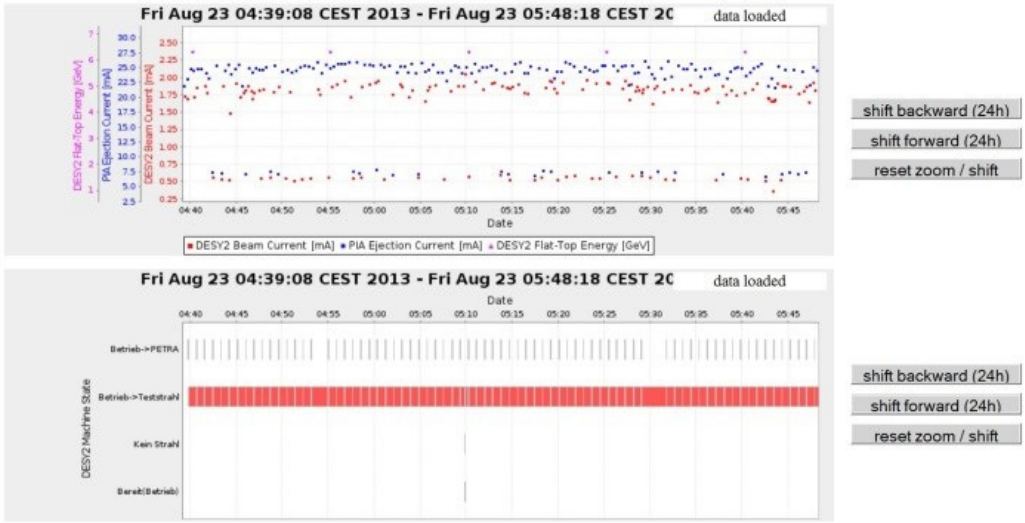


Figure 2: Web2c Archive viewer showing the operations history of the DESY 2 pre-accelerator providing beam mainly for test beam users interleaved with short refilling intervals for the PETRA 3 synchrotron light source running in top-up mode.

## THE Web2cToGo PROJECT

Web2cToGo [3] is a Web2cToolkit-compliant Web-service. Fig. 1 sketches the associated client-server architecture. The Web2cToolkit is a collection of Web-services providing Web-based rich-client control system applications including a synoptic display viewer, an archive viewer (Fig. 2) and others [4] [5].

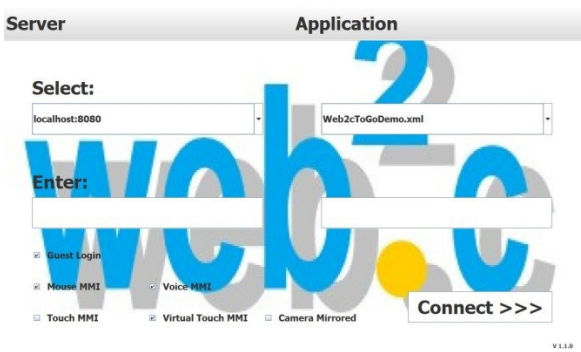


Figure 3: Platform-neutral Web2cToGo application.

Web2cToGo provides a platform-neutral application coming along with implementations for various platforms (Fig. 3). A native mobile app is available for Android, Windows 8 / 8 RT and iOS, a native Java application for all common operating systems having the Java runtime environment installed, and a browser-based Web application running within all major desktop and mobile Web-browsers.

Web2cToGo is a so-called hybrid application. Common to all Web2cToGo implementations is the embedded, platform-independent Web2cToGo Web-Desktop (Fig. 4) coded in HTML, CSS and JavaScript. It provides a framework to launch Web2cToolkit Web-applications (maximum 15), to browse between launched applications or between different pages of the active application, to scroll and zoom the window of the active application, or to interact with the active application.

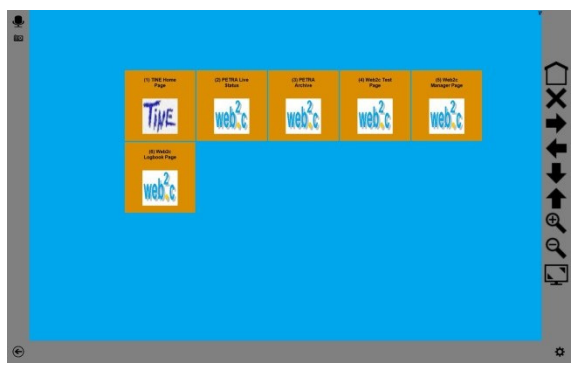

Figure 4: Web2cToGo Web-Desktop showing the Application Explorer (grey left and bottom bars are used for navigation, top-left icons indicate the status of video and audio recording, right-bottom icon opens the toolkit bar at the right border)

### *Web2cToGo MMI Environment*

In addition, Web2cToGo provides an environment to support and test various MMI types including mouse, touch, voice and virtual touch depending on the capabilities of the underlying platform.

Where possible, Web2cToGo uses the PhoneGap (ver. 1.9.0) library [6] to access the platform-dependent and device specific resources such as network, file system, microphone or camera. PhoneGap interfaces with Android, iOS and other mobile operating systems. Unfortunately the features or the performance provided

ISBN 978-3-95450-139-7

by PhoneGap are not sufficient for all uses cases. In particular for the Windows 8 / 8 RT platforms, specific extensions might have to be implemented. The Web2cToGo Java application uses appropriate Java classes only. The browser-based Web2cToGo implementation is restricted to mouse and/or touch interaction only. Platform-independent mouse and touch interaction is provided by HTML.

In the following, this paper discusses the principles, developments, and experiences regarding these interfaces.

### *Touch*

Application control by gestures is supported by all Web2cToGo implementations. The Web2cToGo Web-Desktop catches the events generated by the underlying operation system reacting on a single or multiple touch of a touch-sensitive screen. The event pattern is analysed and assigned to a specific gesture. Finally the action or command associated with this gesture is executed. Implemented touch gestures are (Fig. 5)

- *tap "tile n"* (launches / displays application (with application index n)),
- *swipe "left ↔ right"* (browses to next (previous) application),
- *swipe "top ↔ bottom"* (browses to next (previous) page of the active application),
- *tap "up" ("down", "left", "right")* (scrolls active application page upwards (downwards, towards left, towards right)),
- *swipe "top ↔ bottom"* and *swipe "left ↔ right"* simultaneously (zooms in (out) active application page ),
- *pinch "open"* (displays Application Explorer),
- *pinch "close*" (terminates active application).

The touch gesture recognition is performed within the Web2cToGo Web-Desktop client exclusively and the inherent latency is insignificant for the user.

### *Voice*

Application control by voice is supported by all Web2cToGo implementations except the browser-based version. The Web2cToGo application records periodically a short audio sequence (currently 2.5s long) and uploads the corresponding audio file to the Web2cToGo servlet. Web2cToGo uses the Sphinx-4 (ver. 1.0 beta) speech recognition software [7]. Sphinx-4 is an open-source library package entirely written in Java. The audio input to be analysed by Sphinx-4 has to be provided according to the WAVE/WAV audio file format. According to the embedded language-specific acoustic model the speech recognition software identifies the decomposition into phonemes of the acoustic sequence and finds the best match of recognized phonemes and potential utterances (words, non-linguistic fillers). For example the phoneme's sequence "B R AW Z R AY T" is assigned to the command "Browse right". Finally, the recognized commands are published by the Web2cToGo servlet to be executed within the Web2cToGo Web-Desktop client. Implemented commands are

- *Display "n"* (launches / displays application(with application index n)),
- *Browse left (right)* (browses to next (previous) application),
- *Browse up (down)* (browses to next (previous) page of the active application),
- *Scroll up (down, left, right)* (scrolls active application page upwards (downwards, towards left, towards right)),
- *Zoom in (out)* (zooms in (out) active application page),
- *Scale window* (fits active application page to window)
- *Home page* (displays Application Explorer),
- *Terminate* (terminates active application).

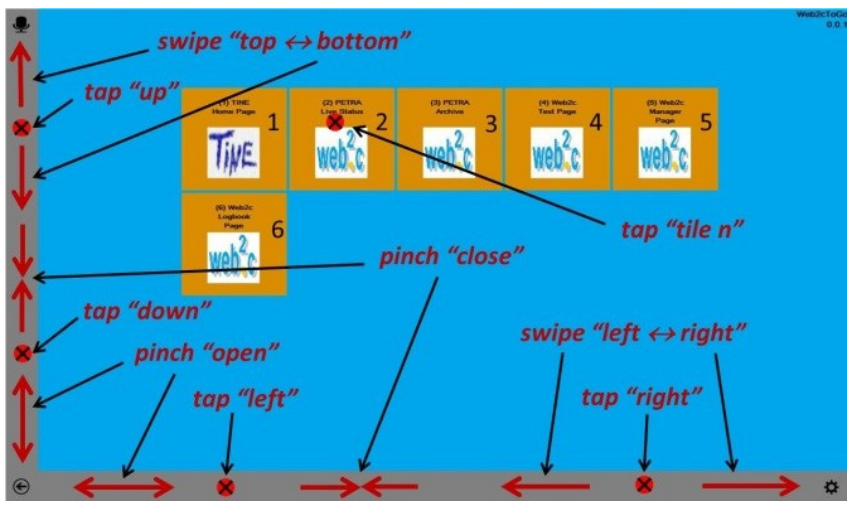


Figure 5: Implemented touch gestures (2D).



ISBN 978-3-95450-139-7

The audio recording quality governs the quality of the speech recognition [3]. Recording humming noise and acoustic artefacts might easily end in useless recognition results.

Web2cToGo speech recognition is based on a common acoustic model not specifically trained to adapt the sound of an individual speaker. The resulting word recognition reliability might not exceed 80% - 90%. In addition, improper speaker-specific pronunciation might also spoil the recognition quality. The speakers should talk fluently and clearly according to the pronunciation expected by the acoustic model embedded.

The concept of local recording and remote recognition results in an over-all latency of a few seconds which is still unsatisfactory for routine use. Local recognition would improve the reaction time at the expense of the platform-independence of the Web2cToGo Web-Desktop client software. Another option might be to replace the unidirectional HTTP communication by bidirectional client-server communication based on the WebSocket protocol [8] but imposing significant security issues.

### *Virtual Touch*

Application control by virtual touch is currently being implemented. The basic idea of virtual touch recognition is a virtual application displayed by a LCD projector and viewed by a camera. Projector and camera are firmly mounted against each other (Fig. 6). A user points towards this projected image and the user's finger position is tracked.

Figure 6: Test setup for virtual touch detection (mini LCD projector on top of a web cam).

The Web2cToGo application periodically (every 1.5s) takes a snapshot and uploads the corresponding image file to the Web2cToGo servlet. The difference image of the current and previous snapshot images is then calculated (Fig. 7). The absolute position of the Web2cToGo Web-Desktop window within the difference image is detected by tracking blinking markers located at the top-left, bottom-left and bottom-right window corners. If a user points towards the projected Web2cToGo application, the corresponding pointer becomes visible and its target position can be calculated with respect to the Web2cToGo Web-Desktop window. Finally, the recognized pointer target position is published by the Web2cToGo servlet to be executed within the Web2cToGo Web-Desktop client.

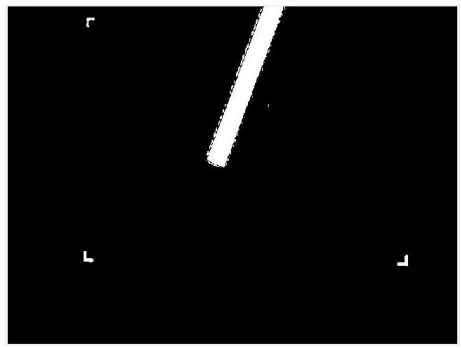

Figure 7: Difference image with markers indicating the absolute position of the Web2cToGo Web-Desktop window within the difference image. A pointer is targeting to the centre of the Web2cToGo Web-Desktop window.

The virtual touch recognition feature of Web2cToGo is inspired by the SixthSense project providing a wearable gestural interface [9]. The development is still in progress.

Similar to the Web2cToGo speech recognition the concept of local recording and remote recognition results in an over-all latency of a few seconds. In that case local recognition seems to be feasible within the framework of HTML5 without breaking the platform independence.

## NEXT STEPS

The Web2cToGo development plan lists a set of extensions, improvements and tests including

- Supporting more MMI types providing motion gesture (3D) and gaze detection,
- Testing Web2cToGo in combination with Google Glass [2],
- Designing and implementing Web2cToGo-compliant apps for specific use-cases, and
- Field tests involving control room operators and service technicians.

## REFERENCES

[1] Microsoft Xbox / Kinect http://www.xbox.com
[2] Google Glass http://www.google.com/glass/start
[3] R. Bacher, "Web2cToGo: Bringing the Web2cToolkit to Mobile Devices", PCaPAC'12, Kolkata, India, December 2012, WEIC01, p. 4 (2012) http://www.JACoW.org
[4] R. Bacher, "Light-Weight Web-based Control Applications with the Web2cToolkit", ICALEPCS'09, Kobe, Japan, October 2009, THP110, p. 889 (2009) http://www.JACoW.org
[5] Web2cToolkit Home http://web2ctoolkit.desy.de
[6] PhoneGap Home http://phonegap.com
[7] Sphinx-4 (Open Source Toolkit for Speech Recognition) Home http://cmusphinx.sourceforge.net/sphinx
[8] WebSocket Protocol Specification http://tools.ietf.org/html/rfc6455
[9] SixthSense Project home http://www.pranavmistry.com/projects/sixthsense

ISBN 978-3-95450-139-7