Web2cToGo: BRINGING THE Web2cToolkit TO MOBILE DEVICES

R. Bacher, DESY, Hamburg, Germany

Abstract

The Web2cToolkit is a collection of Web services. It enables scientists, operators and service technicians to supervise and operate accelerators and beam lines through the World Wide Web. The toolkit includes a synoptic display viewer and editor, an archive viewer, a messenger service, a logbook facility, an administration manager and an HTTP gateway to control systems. Recently, a novel view (Web2cToGo) has been added which is especially designed for mobile devices such as tablet computers or smartphones running iOS, Android or other mobile operation systems. Web2cToGo is a frame which embeds instances of all kinds of Web2c tools. It provides a single-sign-on user authentication and authorization procedure. Web2cToGo supports single- or multiple-touch user gestures and is available as a platform-independent browser-based Web application or as a platform-dependent native app. This paper describes the conceptual design of Web2cToGo and the technologies used behind the scenes as well as the experiences gathered so far. It presents an outlook of on-going developments including user-device interaction based on voice recognition.

INTRODUCTION

The Web2cToolkit [1] [2] is a collection of Web services including

(1) *Web2c Synoptic Display Viewer*: Interactive synoptic live display to visualize and control accelerator or beam line equipment,

(2) *Web2c Archive Viewer*: Web form to request data from a control system archive storage and to display the retrieved data as a chart or table,

(3) *Web2c Messenger*: Interface to E-Mail, SMS and Twitter,

(4) *Web2c Logbook*: electronic logbook with auto-reporting capability,

(5) *Web2c Manager*: administrator's interface to configure and manage the toolkit,

(6) *Web2c Editor*: graphical editor to generate and configure synoptic displays, and

(7) *Web2c Gateway*: application programmer interface (HTTP-gateway) to all implemented control system interfaces.

Web2cToolkit is a framework for Web-based Rich Client Control System Applications. It provides a user-friendly look-and-feel and its usage does not require any specific programming skills. By design, the Web2cToolkit is platform independent. Its services are accessible through the HTTP protocol from every valid network address if not otherwise restricted. A secure single-sign-on user authentication and authorization procedure with encrypted password transmission is provided. Registered and so-called privileged users have more rights compared to ordinary users (read-only permission).

The Web 2.0 paradigms and technologies used include a Web server, a Web browser, HTML (HyperText Markup Language), CSS (Cascading Style Sheets) and AJAX (Asynchronous JavaScript And XML). The interactive graphical user interface pages are running in the client's Web browser. The interface is compatible with all major browser implementations including mobile versions. The Web2cToolkit services are provided by Java servlets running in the Web server's Java container. The communication between client and server is asynchronous. All third-party libraries used by the Web2cToolkit are open-source.

The Web2cToolkit provides interfaces to major accelerator and beam line control systems including TINE [3], DOOCS [4], EPICS [5] and TANGO [6]. The toolkit is capable of receiving and processing JPEG-type video streams.

CONCEPTUAL DESIGN OF Web2cToGo

Web2cToGo is a novel Web2cToolkit service especially designed for mobile devices such as tablet computers or smartphones running iOS, Android or other mobile operation systems. Web2cToGo is a frame which embeds instances of all kinds of Web2cToolkit services.

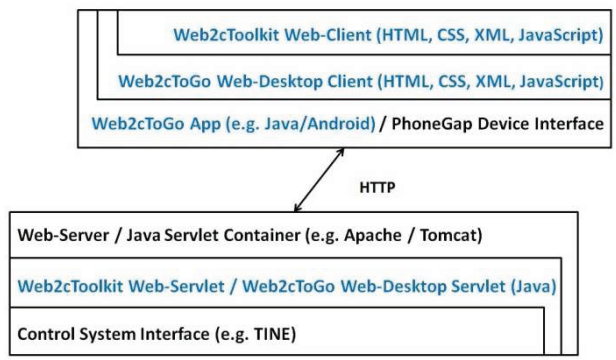

Figure 1: Web2cToGo client-server architecture.

Web-Desktop

The Web2cToGo service consists of the Web2cToGo mobile app and the corresponding Web2cToGo servlet (Fig. 1). By design an app is a platform-dependent application. However, the Web2cToGo app consists only of a few lines of native code creating a component capable of parsing and executing platform-independent HTML- / CSS-tags and JavaScript code snippets implementing the Web2cToGo Web-Desktop Client.

ISBN 978-3-95450-124-3

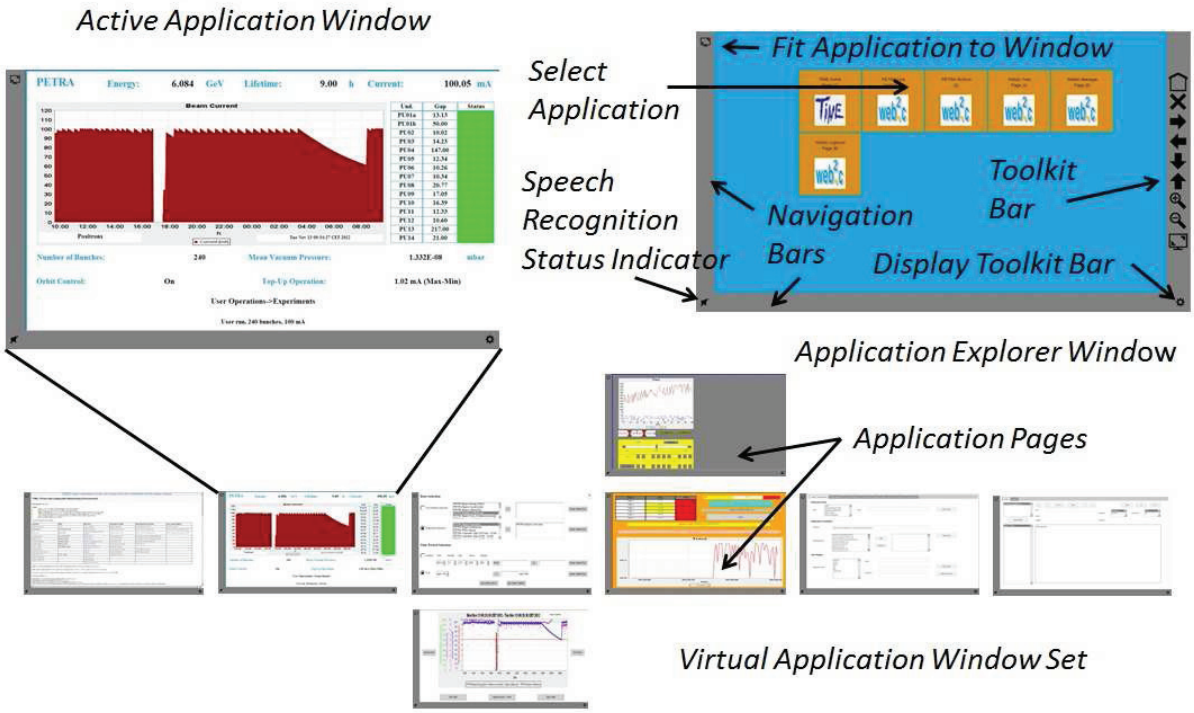

Figure 2: Web2cToGo Web-Desktop.

Access to the platform-dependent and device specific resources such as the network, the file system or a microphone is provided by the PhoneGap (ver. 1.9.0) library [7] interfacing Android, iOS and other mobile operating systems. The Web2cToGo Web-Desktop client is coded in Java within the Google Web Toolkit framework [8] and compiled to HTML- / CSS-tags and JavaScript code which can be handled by all modern Web browsers or similar components ensuring platform independence.

The Web2cToGo Web-Desktop (Fig. 2) provides an Application Explorer which enables the user to select a standard Web2cToolkit Web-Client application by opening a browser window replacing the Application Explorer window*. This browser window requests and displays the corresponding Web2cToolkit application page hosted by the associated Web server. Each application may consist of multiple pages. At most fifteen applications can be started at a single time and kept open in the Virtual Application Window Set.

Navigation and Application Control by Touch Gestures

The user sees only the so-called Active Application Window and navigates between all started applications or application pages and the Application Explorer window using specific single- or multiple-touch gestures. The navigation bars at the left and lower border of the display are sensitive to the user's Web-Desktop navigation actions, only. Implemented touch-gestures are (Fig. 3)

(1) *tap "tile n"* (opens a new application window or displays the selected application window if already opened),

(2) *swipe "left ↔ right"* (switches between applications available in the Virtual Application Window Set),

(3) *swipe "top ↔ bottom"* (switches between application page windows of the active application available in the Virtual Application Window Set),

(4) *tap "up", tap "down", tap "left", tap "right"* (scrolls the active application page window upwards, downwards, towards left, towards right),

(5) *swipe "top ↔ bottom"* and *swipe "left ↔ right"* (zooms the active application page window),

(6) *pinch "open"* (opens the Application Explorer window, and

(7) *pinch "close"* (terminates the active application).

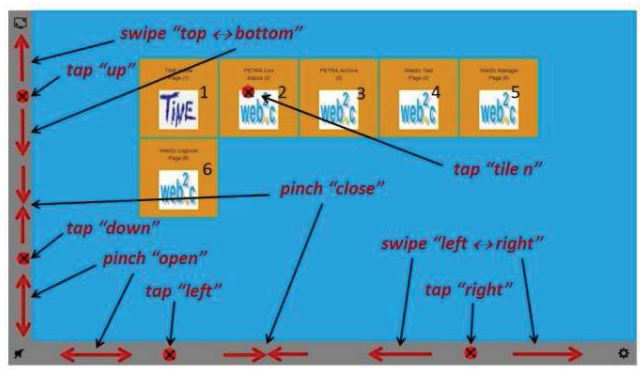

Figure 3: Web2cToGo Web-Desktop navigation with single- and multi-touch gestures (see text).

* The Application Explorer can also launch ordinary Web pages.

ISBN 978-3-95450-124-3

Touch events captured by the browser of the Active Application Window are accounted to and handled exclusively by the corresponding Web2cToolkit Web-Client application. They can be used for application control and will not be passed to the Web2cToGo Web-Desktop for navigation purposes.

Navigation and Application Control by Speech

In a simplified view neglecting the dynamics, speech is an acoustic sequence of utterances (words and non-linguistic fillers) and pauses. Utterances consist of various distinct phonemes. A speech recognition system tries to identify phonemes according to an acoustic model representing the specifics of a language or the sound of an individual speaker. In the next step the system browses a phonetic dictionary listing words and their corresponding decomposition into phonemes in order to find the best match. Optionally, a specific language model constraining the list of words to be recognized and / or defining a grammatical order may guide or simplify the final search.

Web2cToGo uses the Sphinx-4 (ver. 1.0 beta) speech recognition software [9]. Sphinx-4 is an open-source library package entirely written in Java. Audio input to be analysed by Sphinx-4 has to be provided according to the WAVE/WAV audio file format. Web2cToGo speech recognition is based on two acoustic models, US English [10] and German [10].

In Web2cToGo short audio sequences which are only a few seconds long are subsequently recorded by the mobile device and uploaded to the corresponding Web server. If required the uploaded audio file is converted to the proper Sphinx-4 input format using Xuggler (ver. 5.4) [11] providing a Java interface to the FFmpeg (ver. 1.0) [12] media converter. The final audio file is asynchronously analysed by the Sphinx-4 software and the recognized words are finally published by the Web2cToGo servlet to be used by the Web2cToGo Web-Client for navigation or by other Web2cToolkit services for application control purposes.

In our prototype set-up the speech recognition quality turned out to be poor. Less than 20% of the words could be properly recognized. Various reasons for these results have been explored:

(1) An Acer ICONIA TAB A200 [13] tablet running Android (ver. 4.0.3) records the audio files with its internal microphone. The audio recording quality of this device is very limited (dominant humming noise, acoustic artefacts) and is the subject of numerous complaints in the corresponding internet forums.

(2) Android provides audio files with lossy data compression according to the AMR-NB (narrow-band, mono, 8 kHz sampling frequency, 12.2 kbit/s bit rate) or AMR-WB (wide-band, mono, 16 kHz sampling frequency, 6.6 kbit/s bit rate) compression standard. Both formats have to be converted prior to processing by Spinx-4. Dedicated tests with well-known audio test files demonstrate that data compression with AMR-WB and final conversion to WAVE/WAV (mono, 16 kHz sampling rate, 256 kbit/s bit rate) do not spoil the speech recognition quality, while compression with AMR-NB results in a substantial degradation.[†]

(3) Both acoustic models (US English: WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz [10] and German: voxforge_de_sphinx.cd_cont_3000 [10]) suited for 16 kHz data sampling implemented in Web2cToGo provide similar satisfactory speech recognition results, while the acoustic model (US English: WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz [10]) suited for 8 kHz data sampling reduces the recognition quality significantly.

In conclusion, the poor audio recording quality of the mobile device used has been identified as the primary source of degradation in the ultimate speech recognition quality.

PROJECT STATUS AND GOALS

Web2cToGo is an on-going project. The deliverables already implemented include Android 4.x.x Web2cToGo app, platform-independent Web2cToGo Web-Desktop client and servlet, Web2cToGo Web-Desktop navigation by touch gestures and navigation by speech based on the Sphinx-4 compliant acoustic models suited for 16 kHz data sampling.

The goal is to provide additional mobile apps for iOS and Windows RT as well as a Java application suitable for desktop computers. Furthermore, touch gestures dedicated to control Web2cToolkit applications such as history plot zooming in the Web2c Archive Viewer have to be defined and implemented. Finally, handling and reliability of the speech recognition service embedded in Web2cToGo has to be improved and application-specific speech-driven control of the various Web2cToolkit applications has to be implemented.

REFERENCES

[1] 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
[2] Web2cToolkit; http://web2ctoolkit.desy.de
[3] TINE; http://tine.desy.de
[4] DOOCS; http://doocs.desy.de
[5] EPICS; http://www.aps.anl.gov/epics
[6] TANGO; http://www.tango-controls.org
[7] PhoneGap; http://phonegap.com
[8] Google Web Toolkit; https://developers.google.com/web-toolkit
[9] Sphinx-4; http://cmusphinx.sourceforge.net/sphinx
[10] Sphinx-4 Acoustic Models; http://sourceforge.net/projects/cmusphinx/files
[11] Xuggler; http://www.xuggle.com/xuggler
[12] FFmpeg; http://ffmpeg.org
[13] Acer ICONIA TAB A200; http://www.acer

[†] Tests with MPEG-4 audio files (mono, 44.1 kHz sampling rate, 63 kbit/s bit rate) recorded with an iPhone4 running iOS (ver. 6.0) and down-converted to WAVE/WAV files show also promising speech recognition results.

ISBN 978-3-95450-124-3