

ADVANCED DATA EXTRACTION INFRASTRUCTURE FOR MOBILE DEVICES

Mobile user interface and gesture support

Toni Pirhonen

Bachelor's Thesis

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Toni Pirhonen			
Title of Thesis Advanced Data Extraction Infrastructure for mobile devices			
Date	18 March 2014	Pages/Appendices	22
Supervisor(s) Mr. Sami Lahti, Lecturer			
Client Organisation/Partners Karlsruhe Institute for Technology			
<p>Abstract</p> <p>The Advanced Data Extraction Infrastructure (ADEI) is a dynamic web interface for a massive amount of research data. ADEI is used in many scientific researches for collecting measurement data and visualizing it to a web display.</p> <p>ADEI's web user interface (UI) is designed for desktop computers and with the growing amount of hand held devices it was time to implement a mobile UI. The goal of the thesis was to design and implement a mobile UI to the web interface of ADEI. Thesis focused on implementing gesture support and modifying the UI for mobile devices. First the support for touch gestures on mobile devices was inspected. At the time the only mobile browser supporting touch gestures with javascript was safari mobile. It was then decided that the mobile UI support would be limited to Apple mobile devices.</p> <p>As a result the mobile UI was implemented with a custom layout for mobile devices. A custom module for some of the properties in ADEI was also created for easier usage on mobile devices.</p>			
Keywords HTML, PHP, AJAX, XML, XSLT			

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Toni Pirhonen	
Työn nimi Advanced Data Extraction Infrastructure for mobile devices	
Päiväys 18.3.2014	Sivumäärä/Liitteet 22
Ohjaaja(t) Lehtori Sami Lahti	
Toimeksiantaja/Yhteistyökumppani(t) Karlsruhe Institute for Technology	
Tiivistelmä <p>ADEI on kehitetty tarjoamaan datan analysointia, tallennusta ja graafista visualisointia erilaisten tieteellisten tutkimuksien parissa työskenteleville. Datan visualisointi tapahtuu ADEI:n omassa webikäyttöliittymässä, joka on pääasiallisesti suunniteltu pöytätietokoneille. Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa mobiilikäyttöliittymä ADEI:n oman webikäyttöliittymän rinnalle.</p> <p>Työn alkoi mobiiliselainten tutkimisella. Tutkimuksen tuloksena saatiin selville että työn toteutus- hetkellä ainoa javascript kosketustuen omaava mobiiliselain oli Applen mobiililaitteiden käyttämä Safari. Tällöin päätettiin että mobiilituki rajoitetaan Applen mobiililaitteisiin.</p> <p>Mobiililaitteille tarkoitettu käyttöliittymä toteutettiin käyttäen yleisimpiä webohjelmointikieliä. ADEI:n lisättiin myös uusi moduuli datalähteiden valintaa varten. ADEI käyttää tutkimustulosten visualisointiin kuvaajaa, johon lisättiin mobiililaitteille kosketustuki.</p> <p>Työn tuloksena saatiin ADEI:n useimpia ominaisuuksia käyttävä mobiilikäyttöliittymä. Mobiili käyttöliittymä kykenee ADEI:n perustoimintoihin. Datan visualisoinnin lisäksi uutena ominaisuutena lisättiin mahdollisuus lähettää kuvaaja ja siihen liittyvä sensoridata sähköpostin välityksellä haluttuun sähköpostiosoitteeseen.</p>	
Avainsanat HTML, PHP, AJAX, XML, XSLT	

TABLE OF CONTENTS

1	INTRODUCTION	7
2	TOOLS AND TECHNIQUES USED	8
2.1	Development tools	8
2.2	HTML	8
2.3	CSS	8
2.4	XML, XSLT	9
2.5	PHP	10
2.6	AJAX.....	10
3	ADEI.....	11
3.1	Introduction to ADEI 's web interface	11
3.2	ADEI Architecture	12
4	DESIGNING	13
4.1	Objective	13
4.2	Layout.....	13
5	PROGRAMMING	16
5.1	Functionality of the settings module	16
5.1.1	Creating a new page	17
5.1.2	List service	17
5.1.3	Search service	18
5.2	Graph functionality	18
5.2.1	Handling gestures	18
5.2.2	Overlapping controls for graph	19
6	TESTING	21
7	CONCLUSION	22

ABBREVIATIONS AND NOTIONS

ADEI	Advanced Data Extraction Infrastructure
KIT	Karlsruhe Institution for Technology
IPE	Institute for Data Processing and Electronics
KATRIN	Karlsruhe Tritium Neutrino Experiment
UI	User Interface
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations
PHP	PHP: Hypertext Processor
AJAX	Asynchronous Javascript and XML

1 INTRODUCTION

This thesis is about designing and implementing a mobile user interface to the Advanced Data Extraction Infrastructure (ADEI) web interface. Advanced Data Extraction Infrastructure has been developed to provide data exploration capabilities to a broad range of physical experiments dealing with time series. It utilizes a highly modular architecture consisting of backend and frontend parts communicating over the HTTP protocol using AJAX and XML. (Chilingaryan S., Beglarian A., Kopmann A. and Vöcking S. 2010)

ADEI is licensed under the GNU General Public License and uses only free open source technologies. It is currently implemented at slow control systems of Karlsruhe Tritium Neutrino experiment (KATRIN) and Test Facility for Fusion Magnets (TOSKA). There is also an experimental implementation providing access to the data collected at Aragats Space Environmental Center. (Chilingaryan S., Beglarian A., Kopmann A. and Vöcking S. 2010)

The ADEI web frontend is designed in the same manner as Google Maps interface. The navigation through the data can be handled using mouse or certain hot keys. Some of the navigation functionality will be implemented during the course of this project to support Apple mobile devices.

Thesis was made in collaboration with Karlsruhe Institution for Technology (KIT). KIT bundles two institutions: A university of the state of Baden-Wuerttemberg and a large-scale research institution of the Helmholtz Association. KIT operates under three fields of action, research, teaching and innovation.

2 TOOLS AND TECHNIQUES USED

In this chapter the tools and techniques used in creating the mobile interface are explained shortly. First the development tools and environments are explained. Then some details on the used programming, markup and style sheet languages are presented.

2.1 Development tools

The development was done with an iMac computer running Mac OS X – Snow Leopard 10.6.3. Smultron 3.6b1, a sophisticated text editor for code was used to create all the code files. Revision control was handled with Bazaar. Bazaar is a distributed version control system. All the functionality was made in a local branch and later merged in to ADEI's main branch. A local copy of ADEI was running on an Open Suse Linux system for testing and Cyberduck 3.5.1, a ftp-client program was used to transfer files to the server. Testing was handled with iPad running iOS 3.2 and iPod running iOS 4.

2.2 HTML

Hyper Text Markup Language (HTML) is a markup language for web pages. HTML consists of tags enclosed in angle brackets. It was used to describe all the elements placed in the UI. Below is an example of a simple html markup.

```
<html>
  <body>
    <h1>Heading tag</h1>
    <p>paragraph</p>
  </body>
</html>
```

2.3 CSS

Cascading Style Sheet (CSS) is a style sheet language for web pages. Style sheets describe stylistic rules for fonts, colors and layout of the page. CSS was used to style the layout and elements for mobile devices.

An example of using CSS styles heet to style element with the id of “maincontainer”.

```
#maincontainer{
    width: 100%;
    height: 100%;
    background: #000000;
}
```

2.4 XML, XSLT

Extensible Markup Language (XML) is a markup language that handles encoding documents in a format that is both human-readable and machine-readable, which is one the reason xml is widely used in different types of applications.

XSLT (extensible stylesheet language transformations) is a stylesheet language for XML. In this thesis XSLT was used to check which data to display and format it correctly to the UI. Example of XSLT used to format an email from generated XML

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="utf-8"/>
  <xsl:template match="result">
    <div class = "emailform">
      <divclass="emailheading"><table
class="headingtable"><tr><td><h1>List
of Sensors</h1></td></tr></table>
      </div>
      <div class="emailcontent"> <table>
        <tr><td>ID</td><td>Name</td></tr>
          <xsl:for-each select="value/item">
            <tr><td><xsl:value-of
select="itemid"/></td><td><xsl:value-of se-lect="itemname"/></td></tr>
          </xsl:for-each><tr><td><button><xsl:attribute
name="onclick">javascript:graph_control.closediv('sensordiv');</xsl:attri
bute>Close</button></td></tr></table></div> </div>
    </xsl:template>
  </xsl:stylesheet>
```

2.5 PHP

PHP is a server-side scripting language, which is used to create dynamic web pages. It can be implemented directly in the html document and the code itself is handled by the server and never displayed to the client. Like html, PHP elements consist of tags enclosed in angle brackets. PHP also includes support for various types of databases like MySQL, SQLite and Oracle.

An example of using PHP inside a HTML document

```
<html>
  <?php $age = 30; ?>
  <p>I am <?php echo $age ?> old.</p>
</html>
```

2.6 AJAX

AJAX (Asynchronous Javascript and XML) is a group of interrelated web development methods. Ajax is used on the client-side to create responsive, asynchronous web applications. The goal of Ajax is to bring more interactivity, speed and usability to web pages. Instead of handling all requests by refreshing the page or redirecting the user to another page, AJAX can also handle requests in the background. The content can be dynamically updated to the page.

An example of using AJAX to send a simple asynchronous GET request.

```
xmlhttp.open("GET","get_data.php",true);
xmlhttp.send();
```

3 ADEI

3.1 Introduction to ADEI's web interface

ADEI's interface is planned mainly for desktop computers so there are many elements visible in the normal view. The header of the page consist of a search field and links to all of the subpages. On the left side of the UI there are two control panels which can be hidden by clicking on them. The Footer is designed to always show what is happening when the user is interacting with the ADEI UI. The content section shows the content according to the page that is selected from the header bar. (FIGURE 1)

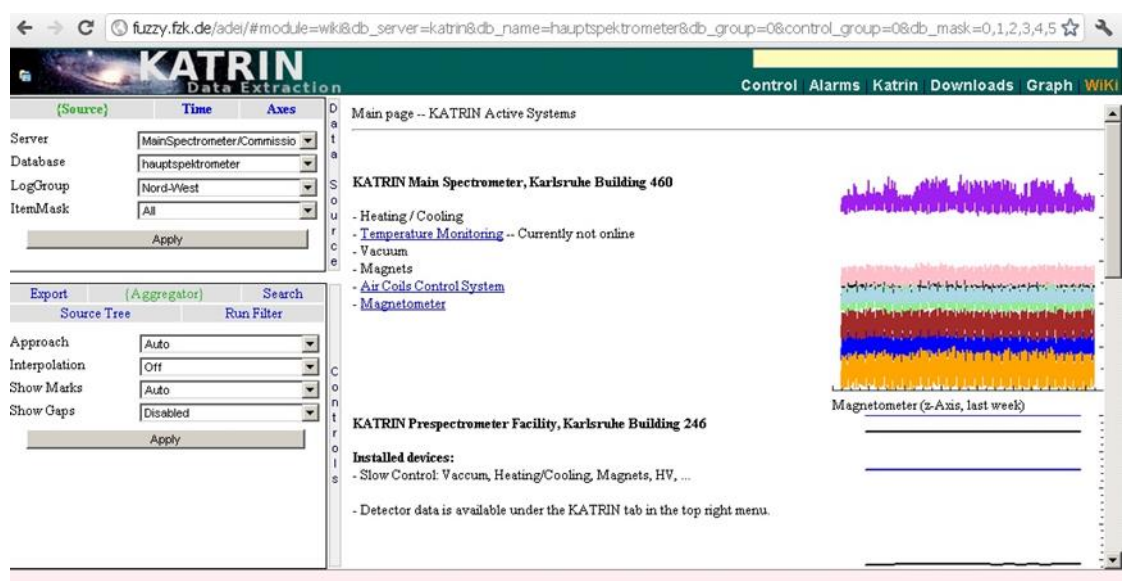


FIGURE 1 ADEI's main view in the KATRIN experiment

3.2 ADEI Architecture

ADEI is designed to work on many different kinds of physical experiments therefore it supports multiple export formats and a direct connectivity to LabVIEW. ADEI also utilizes a highly modular architecture which was beneficial in implementing the mobile UI. The modular design made it possible to implement the mobile UI without increasing the load time of the site or interfering with the main UI. As Figure 2 shows ADEI has 3 main layers in its architecture. Applications, Web Services and Data Sources. Applications describe different use cases for ADEI. Data collected by ADEI can be displayed and exported in various ways. (FIGURE 2)

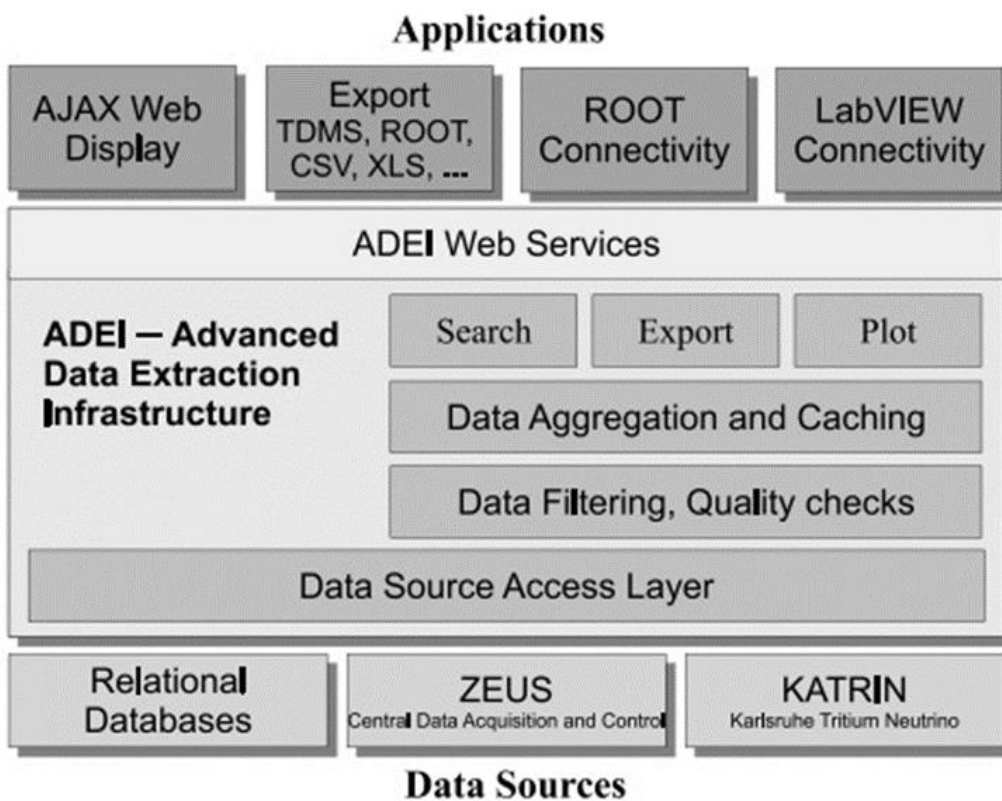


FIGURE 2 Architecture of ADEI (Chilingaryan S., Bedlarian A., Kopmann A. and Vöcking S., 2010)

4 DESIGNING

4.1 Objective

Goal of the project was to create an intuitive mobile web UI for ADEI to be used with iPad, iPod and iPhone. It was defined that the mobile UI has to work independently without interfering with the main ADEI UI for desktop PCs. The Mobile UI was designed to be a easy and accessible way for the users to interact with the research data anytime, everywhere from a mobile device.

4.2 Layout

When designing the layout the small size of handheld device displays had to be kept in mind. ADEI's desktop layout includes two control items labeled Data Source and Controls. Using ADEI's built in properties it was possible to discard those two elements and fetch the needed data to a new module using ADEI's list-service. Also the links from the top of the page desktop version were removed and replaced with a dropdown menu on the right top corner of the page to keep the UI as minimalistic as possible. (FIGURE 3)

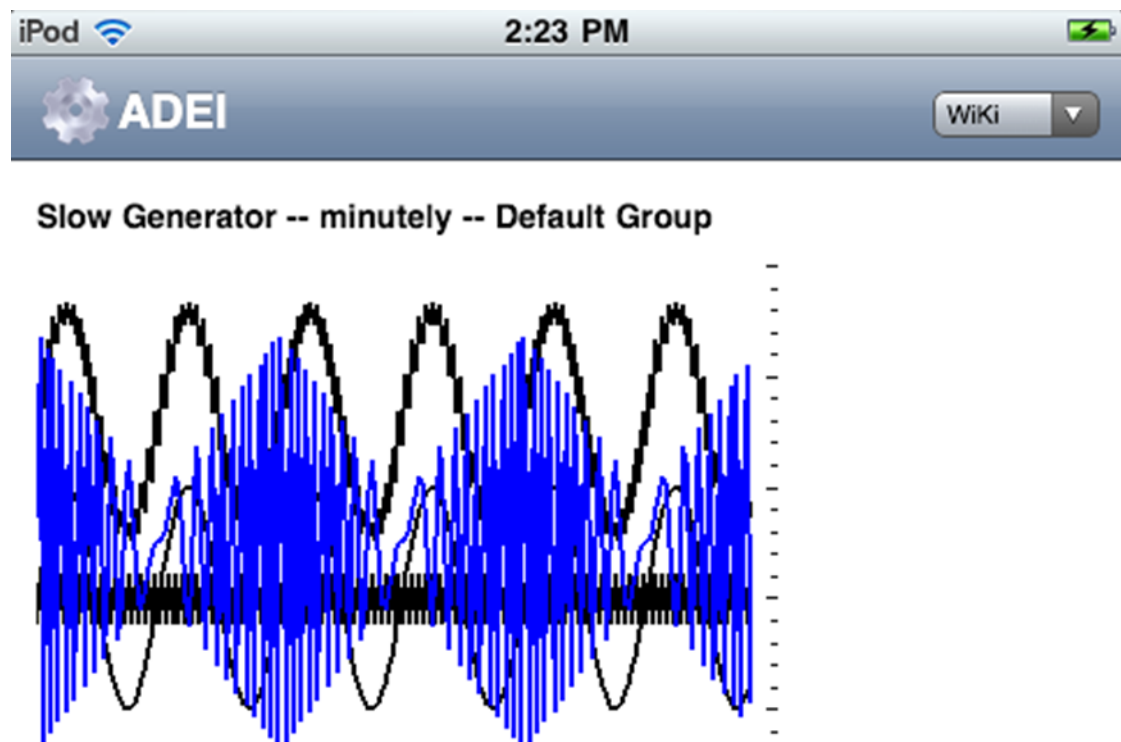


FIGURE 3 ADEI's main view in mobile version

Since the “Data Source” and “Controls” elements were removed from the mobile UI, they were placed as a new settings page behind the cogwheel icon. The controls were made in a waterfall kind of way, which is familiar to iOS users and feasible to use on a mobile device. (FIGURE 4)



FIGURE 4 ADEI’s mobile versions Settings page

On the graph page, new overlaying controls div was implemented for sending the graph and its sensor info via email to a user defined address. Controls were placed inside a transparent div which the user could make visible and hidden with the black and white button on the down right corner of the graph. The basic navigation functionality was also added to the over laying controls as buttons for easy access. Navigation trough the graph data was also implemented with touch gestures. (FIGURE 5)

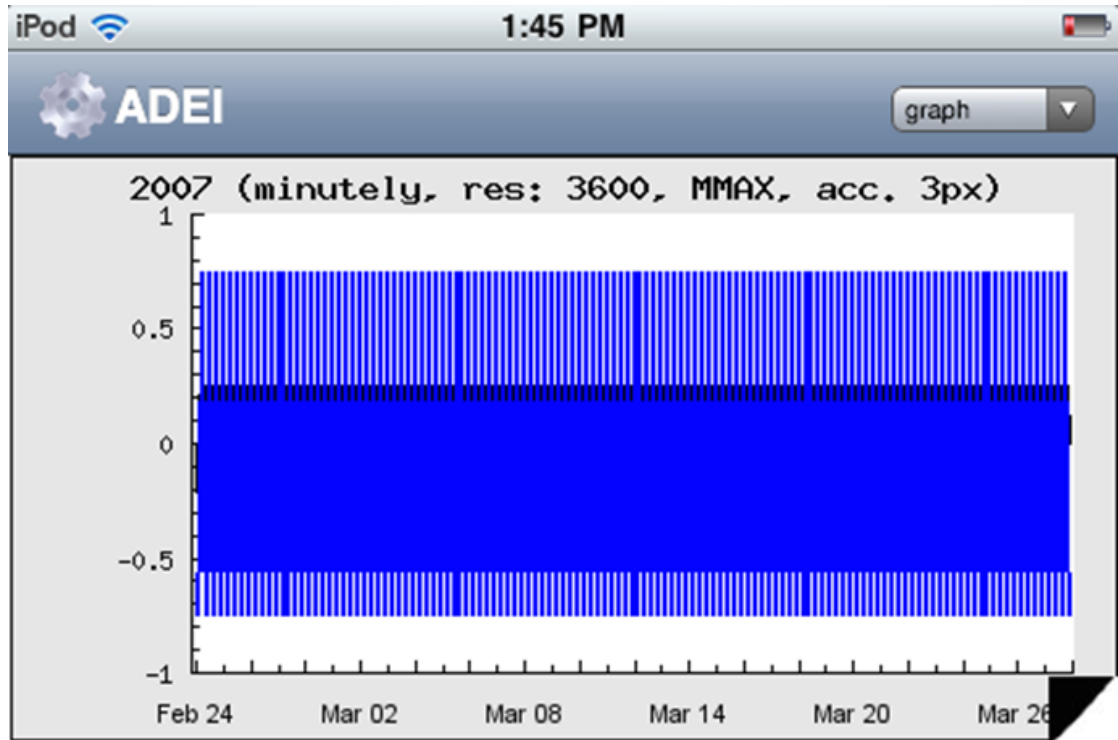


FIGURE 5 Graph page in the mobile ADEI

5 PROGRAMMING

ADEI's user interface is designed to work fluently in desktop browsers. For the UI to work as designed in mobile devices a new module had to be created for Data Source and Controls elements. The desktop version holds the elements on the left side of the UI, in the mobile UI those two elements have been completely discarded and replaced with a new page named Settings which uses the ADEI's built in services to get the necessary data for the UI.

The graph usage in the desktop version of ADEI is done by mouse or by predefined hot keys. In the mobile version touch gestures were implemented for feasible use. Option was also added for sending the current graph and additional information via email.

In this chapter, all the programmed functionalities onto the mobile UI will be described. Starting from the creation of the custom module named "Settings". Then the graph functionality for touch devices including new over laying controls for the graph page will be explained.

5.1 Functionality of the settings module

The settings module uses ADEI's built in services to fetch data. ADEI utilizes many services which can be used by HTTP GET/POST requests (FIGURE 6). The settings module uses mostly list service to list all the needed query parameters to the user and generate a graph according to the parameters selected by the user.

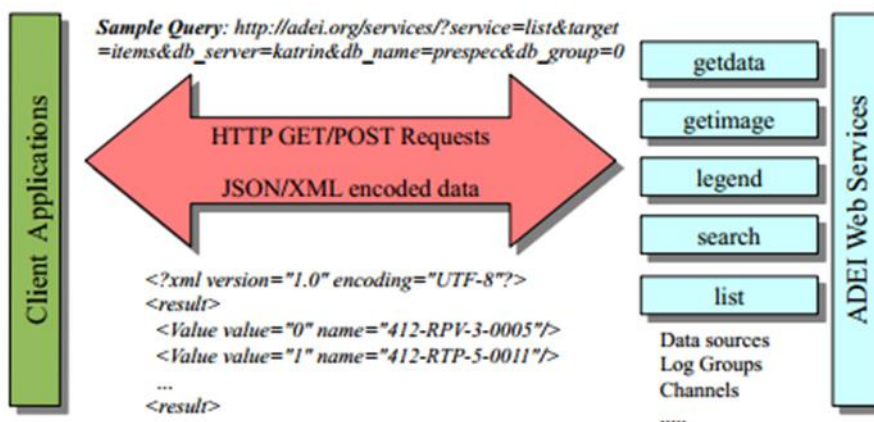


FIGURE 6 ADEI's Web Services (Chilingaryan S., Kopmann A. 2011)

One of the query parameters is time interval, which is an optional parameter that can be used to fetch the research results in the user defined time interval. In the mobile UI time interval can be selected by predefined values or a custom time range. The desktop version of ADEI uses a calendar view for selecting the time interval. In the mobile UI the custom time range was done with a 3rd party javascript library called Spinningwheel.js. Spinningwheel.js library emulates the functionality of a iOS's default selector. (FIGURE 7)



FIGURE 7 Spinningwheel.js example

5.1.1 Creating a new page

When creating a new page ADEI requires few files to be created. First a new service had to be created. This service acts as a small application providing data in the XML format. Then a new module was created. In the module declaration the name “Settings” was defined and the type of the module was defined as a “CONTROL_MODULE_TYPE”. By defining the module type as such, ADEI provides access to the control information needed for the settings page. The module uses the settings service for generating XML data and a XSLT style sheet to update the information on the page.

5.1.2 List service

The list service is used to list different aspects of the ADEI server configuration. The settings module uses list service to fetch all the data sources provided by ADEI. Data sources contain all the servers and their properties. All the properties are dependent on each other, so the mobile UI uses a waterfall approach to list the data to the user. When the user navigates to the settings page a static site listing the two replaced UI

elements “Data Source” and “Controls” are displayed as a simple list. The user can then navigate step-by-step selecting a group of properties which then are used to generate a graph.

5.1.3 Search service

Search service performs searches for various data in ADEI and its source databases. In the mobile UI search is implemented as a single search field in the settings module under the “Controls” page. The search field takes a single string as a search parameter and uses ADEI’s search service to perform the search and lists the result as links which can be used to generate a graph.

5.2 Graph functionality

The graph page consists of the graph, over laying controls for the graph and a header bar for navigation. For users to use the graphs functionality with mobile devices, touch gestures were implemented for controls. The functionality included some of the desktop version’s functions like zooming and moving through the timeline. A feature for overlapping controls was also added which contains the same functions as swipe gestures. An additional property for emailing the graph and i’s sensor list to a colleague was implemented so that the user could receive the graph as a picture and also get all the channel data associated to the graph. In the desktop version of the ADEI UI, the user can select an area with the mouse and select show info which would then load the channel data of the selected area to the left side control panel.

5.2.1 Handling gestures

iOS comes with a web browser based on WebKit, which includes handling of touch gestures. Touch gestures can be handled with JavaScript, like mouse events. Since at the time of creating this project no other mobile browser had gesture support, it was decided that only Apple’s mobile devices will be supported.

Navigation of x-axis was handled with swipe gestures. The user can move in the time axis to the desired direction by swiping across the screen. In Figure 7, the swipe initiates a call to ADEI’s web service to draw a new graph from a time interval with a predefined jump in time. (FIGURE 8)

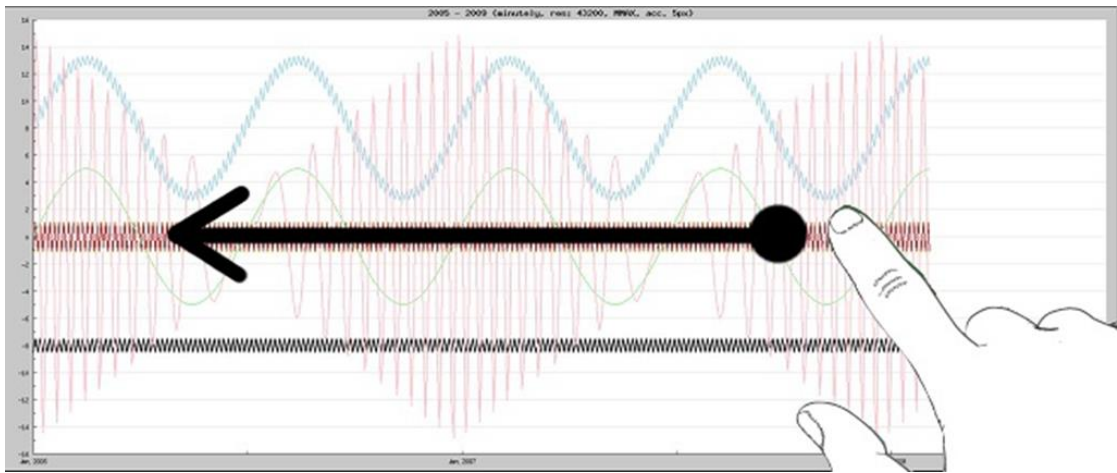


FIGURE 8 Moving forward in the x-axis with the swipe gesture

Zooming in or out of the graph was handled with pinch gesture. Zooming in creates a call to ADEI's web service which zooms both the x and y axis using the center of each axis as a reference. In the Desktop version users can use the same function with the mouse scroll or using the keyboard. (FIGURE 9)

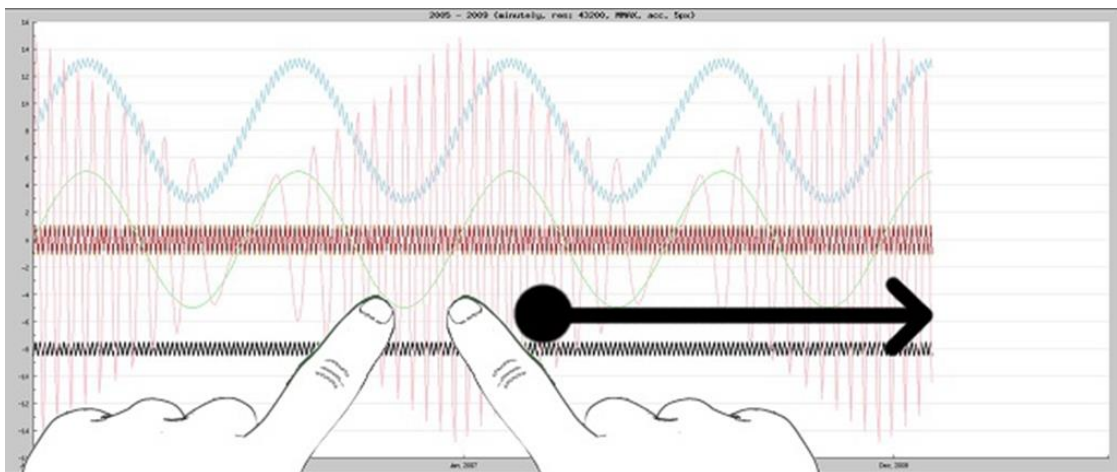


FIGURE 9 Zooming in with the pinch gesture

5.2.2 Overlapping controls for graph

After the implementation of the navigation based on touch gestures to the graph, it was decided that overlapping controls should also be added as an alternative navigation. A see through section was placed on top of the graph with few functional buttons for the navigation and a possibility to send the graph as an email to a user defined address. The navigation buttons included buttons for zooming and navigational buttons for moving in any direction of the graph's axis. For the email functionality a PHP library caller PHPMailer v.5.0 was used. PHPMailer is an open source library for sending html formatted emails. First the settings module fetches all the sensor names

associated with the graph by using ADEI's web services and then formats them in a nice HTML format. The formatted HTML is used as the mail's message. A picture of the currently displayed graph is also generated by using ADEI's web service and then added as an attachment file to the mail. The mail property gives ADEI mobile UI more usability, since any one can easily browse through the data and send the information to any mail for further exploration.

6 TESTING

For testing, two Apple mobile products were used, an iPod and an iPad. Both devices were running on iOS 4 and the default browser Safari was equipped with Webkit 553.17.9. Testing was conducted as an ongoing UI usability and functionality test during the whole projects lifecycle.

The mobile user interface was developed in parallel with other ADEI developers and was merged to the main branch of ADEI in the end of development. After merging the mobile ADEI UI to the Desktop browser version, all the supported internet browsers were also tested. Testing was conducted to verify that mobile UI does not affect in any way to the normal behaviour of ADEI. The changes made in the main ADEI were not conflicting with the project.

No further testing was done at this point since the time allocated for the project was finished.

7 CONCLUSION

The goal of the thesis was to implement a mobile user interface to the ADEI web display. A mobile UI was designed and most of the planned functionalities were added. At the end of the project we noticed that there was not enough time to implement “Legend” property to the graph. That was the only property that was left out.

Working with ADEI was challenging, because of the scale and size of the ADEI project. It took me a while to understand the modular structure and coding practices used in ADEI. After few weeks of trial and error I started to get the hang of some parts of the ADEI programming practices. After that every step of the work was a bit easier. If I were to go to work on a similar project in the future I would prepare better and get familiar with the project beforehand.

The end result was a working mobile UI for Apple mobile devices. In the future development support for a larger quantity of different devices should be added. After the thesis was completed many other mobile browsers have much wider support for mobile web development.

Working in KIT IPE was an eye opening experience into the world of exploring scientists and researchers. I got to develop my programming skills and I gained a lot of knowledge from my colleagues. The time spent on this project was exiting and made me think about my future as an engineer.

REFERENCES

The ADEI Project 2012 A publically visible implementation of the ADEI supporting the KATRIN project at KIT [online] [Accessed 12 December 2013]

Available from: <http://katrin.kit.edu/adei/>

The ADEI Project 2012 The Advanced Data Extraction Infrastructure project's Trac-page [online] [Accessed 17 December 2013]

Available from: <http://adei.info>

Chilingaryan S., Beglarian A., Kopmann A. and Vöcking S. 2010 Advanced Data Extraction Infrastructure: Web based system for management of time series data [online] [Accessed 17 December 2013]

Available from: <http://iopscience.iop.org/1742-6596/219/4/042034>

Matteo Spinelli, 2009 Spinningwheel.js javascript library home page. [online] [Accessed 17 March 2014]

Available from: <http://cubiq.org/spinning-wheel-on-webkit-for-iphone-ipod-touch>