Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 653–660

Procedia Computer Science

The 8th International Conference on Mobile Web Information Systems (MobiWIS)

mCHOIS: An Application of Mobile Technology for Childhood Obesity Surveillance

Arun K. Datta[a,b,]*, Andi Sumargo[a], Victoria Jackson[c], Pradip P. Dey[b]

[a]NU Community Research Institute (NUCRI), and the [b]School of Engineering & Technology (SOET), National University, 11355 N. Torrey Pines, La Jolla, CA 92037, USA

[b]Illinois Department of Human Services, School Health Program, 823 E. Monroe, Springfield, Illinois 62701, USA

Abstract

New mobile computing devices including smartphones and tablet computers have emerged to facilitate data collection in real-time at the point-of-care. Earlier, we developed a web based Childhood Obesity Informatics System (CHOIS) and deployed it for obesity surveillance by the Illinois Department of Public Health (IDPH). In the process, a school nurse collects data on an individual's height and weight for determining Body-Mass-Index (BMI), which is conventionally used for determining at-risk and obese patients. However, this process is often limited by the internet access at the site. This paper describes a solution by demonstrating a smartphone-based mobile application, mCHOIS. The application developed in this project enables a field worker to input or modify the data and store it locally in the phone. Once internet connection is available either through the broadband or through the built-in wifi, data can be sent to the remote database of CHOIS. Updating the data and visualization of the report are also available through the phone's browser. This application has been successfully field tested and is now under deployment for use by the Illinois Department of Human Services (IDHS) for its School Health Program.

Keywords: Mobile Technology; Android; Smartphone; Body-Mass-Index (BMI); Obesity; Point-of-Care (POC); HTML5.

1. Introduction

Advances in the area of mobile and wireless communication coupled with more robust software development for delivery of contents allow the design and development of new patient-centric models [1, 2] for better personalized healthcare services. Mobile devices, such as smartphones and tablets, overcome the limitation of stationary computer systems and are enabling the healthcare field workers to access the data anytime, anywhere. Wide adoption of mobile computing technologies has potentially improved information access for healthcare, enhanced the workflow, improved the quality through better access to relevant data and has promoted evidence-based practice to make informed and effective decisions at the point-of-care [3]. The use of mobile technology in healthcare affords numerous methodological advantages over traditional methods, including the ability to capture time-intensive longitudinal data, date- and time-stamped data, and the potential for personalizing information in

* Corresponding author (arundatta.info). Tel.: 001-858-642-8535; fax: 001-858-642-8769.
E-mail address: adatta@nu.edu

doi:10.1016/j.procs.2011.07.085

real-time. Moreover, the ubiquitous nature of these mobile devices in daily life has created opportunities for improved information access for research applications that were not previously possible, such as, simultaneously assessing behavioral, physiological, and psychological states of an obese patient in real-time. Thus, mobile technology application in healthcare offers great potential to transform medical research and improve modern medicinal practice at the individual level[a] [4, 5]. Our on-going developmental effort on Childhood Obesity Informatics System (CHOIS) and its mobile applications was initiated with that objective [6, 7].

CHOIS has been developed for obesity surveillance [6, 7] and tested in Illinois by IDPH. Obesity, particularly among children, is a national crisis in US [8, 9]. A task-force has been set by President Obama to control this epidemic (White house, Office of the Press Secretary, February 09, 2010). Efforts have been initiated in several states including in Illinois to survey students routinely and identify at-risk and obese children for intervention program with proper diet and exercise [10, 11]. The American Academy of Pediatrics (AAP) recommends that the Body-Mass-Index (BMI) should be measured on all youth as part of normal health supervision [10]. BMI, which is easy to measure and correlates with body fat, assesses the weight status of an individual to identify those at-risk and obese [12]. Automatic computation of BMI, BMI percentile and the risk of obesity alert are embedded into CHOIS. In the process, the School Nurse collects demographic data along with the students' physical condition including height and weight and enters into the web based CHOIS. However, this process is limited by the internet access at the site. To overcome this problem, we have developed a smartphone-based mobile application, termed mCHOIS, for using in the field. This application enhances the current system (CHOIS) with the power of mobile technology and allows the health professionals to gather the client data and save it locally in the phone. When the internet access is available either through broadband or built-in wifi of the phone, these data can be transmitted to the remote database of CHOIS. This application also enables a user to update the data and visualize the report in real-time through built-in browser of the phone.

2. Materials and Methods

Smartphones, available in the US market, come with various kinds of operating systems, namely, IOS (Apple), Android (Google), Symbian (Nokia), Mobile Windows (Microsoft) and Palm OS (Research in Motion). Presently, cross-platform development of mobile application is not technologically feasible (see below). We used Google phone G1, an Android-based phone, for mCHOIS application development. Android is a customizable software platform and operating system for mobile devices [13, 14]. It was developed by Google and later by Open Handset Alliance (OHA)[b]. Google and OHA made Android available as open source software for mobile application developments. Android delivers a complete set of software for mobile devices: operating system, middleware and key mobile applications[c]. Android relies on Linux OS for core system services such as security, memory management, process management, network stack, and driver model. For mobile application development, it has several advantages over other systems, as for example, iPhone, which is a popular smartphone in the market. iPhone application development uses Objective-C [15], which is confined to the Mac World. On the other hand, Android allows developers to write codes in a Java-like language that utilizes Java libraries[b]. Android application, on the other hand, can be developed on any platforms, namely, Windows, Linux and Mac OSX. Android also contains a rich set of Application Programming Interfaces (APIs) that allows third-party developers to develop any applications for the mobile industries. In addition, this software system comes with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. Such applications can easily be integrated with a third party software system, particularly from Open Source. Its browser component is also flexible to accommodate plug-ins, unlike the iPhone, which does not allow Flash Player plug-in (Adobe/Macromedia), for example. Moreover, Android-based Google phone offers all the Google tools, such as Google map, IM for video chat, etc., that can be valuable for location based services at the point-of-care (POC). All these features and more make Android-based application development cost-effective and an attractive choice over other systems.

---

[a] http://obssr.od.nih.gov/training_and_education/mhealth/index.aspx
[b] http://www.openhandsetalliance.com/press_102108.html
[c] http://developer.android.com/index.html

3. Design & Development of mCHOIS

CHOIS and its mobile application have been designed following standard design principles and processes [6,16-18]. Development of this mobile application was done with the following modules:

3.1. Interface: The application in the mobile device provides a GUI to the user with options such as: Data *Input,* Data *Update*, *Send* data and *Visualize* the report (Figure 1). The layout of each front-end-interface/screen is created in XML. We used the Android plug-in for Eclipse (IBM), which is also an Open Source Development Platform that can

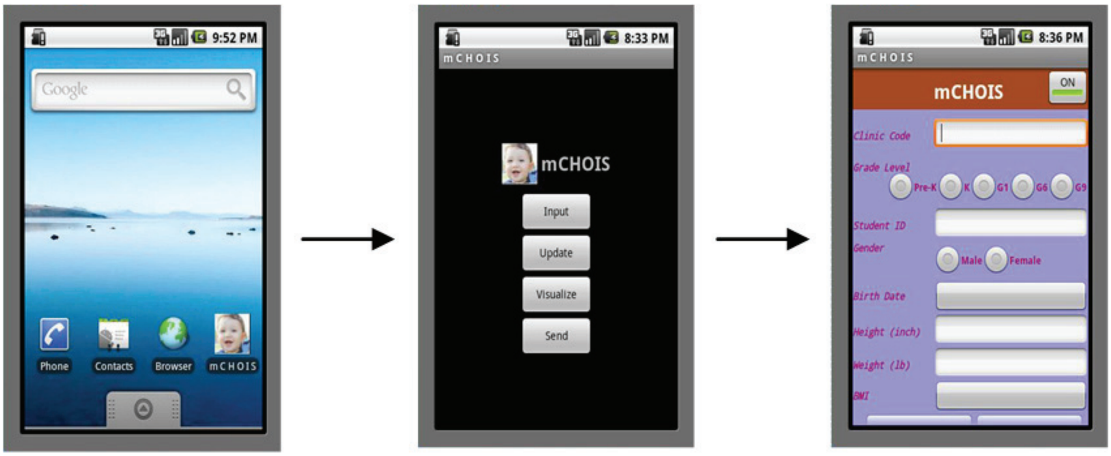

Figure 1. Client view. The client view on the phone is described from left to right as follows: a) the first screen represents the general welcome screen on a Google's phone G1. There is an icon installed on the screen for easy recognition and access to the mCHOIS application developed; b) when the application is started by clicking on the icon, a welcome screen of the application appears which has several options to choose from, namely: Data 'input', Data 'update', 'Send' data and 'Visualize'; c) The input form on the screen is for data input which has text fields and calendar pop up screen to enter date and a button to save data.

be used to write Java, PHP, C/C++ and more languages. The Android uses Java and a plug-in is available to make it easier to write codes for the Android. A directory is also available when using Android plug-in in *Eclipse*. One of those is *Res* directory. *Res* directory contains layout and value subdirectory. The XML that governs the 'User Interface' are located in these subdirectories. *Buttons*, *Edit Text*, *Label*, and *Alignment* (according to the pixel) features are created in this layout. Android platform defines a class 'Activity'. Activities represent the application screens in Android. Activity calls the *setContentView* method to assign the layout that was defined in XML layout. All the activities have to extend from 'Activity' class. Switching between the activities or working on a different activity from the existing activity in the application is done using the 'Activity' *intent*.

The application developed for mCHOIS has the following processes: From the application main menu, user can select a task to perform. When inserting a data input in the form of the application, user can collect the information, enter it into the form, and save it locally in the SQLite database of the Google phone. Another task is to update a data that has been previously entered. Selecting the data update will enable a user to see all the entries that have been saved locally, select the target entry, modify the data, and then save it back to the local database. After the process of data input or data update is finished, user may send the data to the remote server. By selecting the 'Send' button, the application will retrieve all locally stored data, generate the XML formatted document(s), and send it to the remote server of CHOIS over the internet.

3.2 Application Architecture

There are several activities for the mCHOIS application development. The activities and the corresponding codes are given below for clarity:

The file 'MchoisActivity.java' is the application's welcome screen and the only entry point to the mobile application. MchoisActivity uses the main_menu.xml to describe its layout. This is done by calling setContentView() method as:

```
@Override
public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main_menu);
```

MchoisActivity's main function is to start activities depending on what task needs to perform. The welcome screen let a user choose what action needs to perform, such as, data input, data update or sending data. Once an action is selected, then the new activity is called using startActivity(intent) or startActivityForResult(intent, action). InputActivity.java contains the form for data input. Once data is entered, the application will save the data into local database. This is being done in saveState() method. The data is passed to the DbAdapter class as follows:

```
if (RowID == null){
   Long id = mDBHelper.insertData(temp_scode,
        temp_grade, temp_studentid, temp_gender,
        temp_bday , temp_height, temp_weight,
        temp_bmi, temp_today);
   if (id > 0){
      RowID = id;
   }
}
else {
   mDBHelper.updateItem(RowID,temp_scode,
        temp_grade, temp_studentid, temp_gender,
        temp_bday , temp_height, temp_weight,
        temp_bmi, temp_today);
}
```

The file AndroidManifest.xml contains the information needed by the Android to run an application. It listed all activities involved in the application along with its *intent*, if defined. Since the application will open a connection through the internet, it needs permission to access the internet executed by the following codes:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
  android:versionCode="1"
  android:versionName="1.0" package="nucri.mchois">
  <application android:icon="@drawable/child"
     android:label="@string/app_name">
     <activity android:name=".MchoisActivity"
        android:label="@string/app_name">
        <intent-filter>
           <action android:name="android.intent.action.MAIN"/>
           <category android:name=
               "android.intent.category.LAUNCHER"/>
        </intent-filter>
     </activity>
     <activity android:name="InputActivity"></activity>
     <activity android:name="SendActivity"></activity>
     <activity android:name="Update"></activity>
     <activity android:name="Visualize"></activity>
  </application>
  <uses-sdk android:minSdkVersion="4" />
  <uses-permission android:name="android.permission.INTERNET">
  </uses-permission>
</ manifest >
```

3.3. SQLite database: In the Android Operating System, the SQLite database management system is used to manage its own database. The input/update is directly reflected in the database when user performs the specified action. The actions performed in the application with the database are: Create a Database (this is done just once), Open the Database, Create a Table, Insert some Datasets, and Query the Datasets. SQLiteDatabase class provides the developer with the necessary methods to work with the Android's phone-based local database. DbAdapter.java hides the implementation on how to work with the database directly as follows:

```
private static final String DATABASE_CREATE =
    "create table mchois_table ( _id integer "
    + "primary key autoincrement, "
    + "school_code text not null, "
    + "grade_level text not null, "
    + "student_id text not null, "
    + "gender text not null, "
    + "bday text not null, height text not null, "
    + "weight text not null, bmi text not null, "
    + "date_recorded text not null"
    + ");";
```

First, a string containing SQL statement was written to create a new database. Then the execSQL() method from the SQLiteDatabase class was used to run this statement. But the database is not actually created or opened until one of the getWritableDatabase() or getReadableDatabase() is called as shown below:

```
@Override
public void onCreate(SQLiteDatabase db){
        db.execSQL(DATABASE_CREATE);
}

public DbAdapter open() throws SQLException {
        mDbHelper = new DatabaseHelper(mCtx);
        mDb = mDbHelper.getWritableDatabase();
        return this;
}
```

DbAdapter also provides methods to create new item, delete or update an existing item. 'Update' method will try to find an item that has the required *rowId* and update the content using the value provided in ContentValues object. SendActivity.java will be called by the user, when internet connection is available. Its main functions are to retrieve data from local database, put the data in XML document, and send it to the server. The code below generates a Document object containing the data:

```
DocumentBuilder builder =
   DocumentBuilderFactory.newInstance().newDocumentBuilder();

   document = builder.newDocument();
   Element root = document.createElement("bmidata");
   document.appendChild(root);

   Element item = null;
   notesCursor.moveToFirst();

   for (int i=0;i < count;i++){
        item = document.createElement("student");
        item.setAttribute("school_code",notesCursor.getString(1));
        item.setAttribute("grade_level",notesCursor.getString(2));
        item.setAttribute("student_id", notesCursor.getString(3));
        item.setAttribute("gender",   notesCursor.getString(4));
        item.setAttribute("bday", notesCursor.getString(5));
        item.setAttribute("height", notesCursor.getString(6));
        item.setAttribute("weight", notesCursor.getString(7));
        item.setAttribute("bmi", notesCursor.getString(8));
        item.setAttribute("date_recorded",notesCursor.getString(9));
        root.appendChild(item);
        notesCursor.moveToNext();
   }
```

Then the getXml() method transforms the document into a XML-formatted String. This String is sent to the server using a POST method as it is defined in WebInfoSubmitter class as shown by the following:

```
WebInfoSubmitter wis = new
WebInfoSubmitter(requestPath);
response = wis.sendXMLData(xmlInfo);
```

3.4. Transfer Protocol: Data that is gathered from the local SQLite database is processed into XML format and posted to the remote server as a HTTP post after sending the request.

3.5. Remote Server: The string received from the phone by the remote server is parsed into the XML formatted string to read the data. The data parsed is then saved in the remote MySQL database of CHOIS [6], which can be visualized in real-time over the internet through the built-in browser of the phone.

3.6. Server side Programming

XML parsing: The server has to parse the XML formatted string received from the client/Mobile to read the data, validate and store it in the database. This has been executed with the following codes:

```
//concatenate xml declaration with the xml elements sent from android
       $xmldata = $xmltop . $xmlInfo;
//parse xml
       $doc= new DOMDocument();
if (!$doc->loadXML($xmldata))
{
echo ("Error in XML document");
}
//xpath to target student element

$domXPath = new DOMXPath($doc);
//create the attribute name array for easy value retrieval
$xmlAttrArray = array('school_code', 'grade_level', 'student_id', 'gender',
                  'bday', 'height', 'weight', 'bmi', 'date_recorded');

  //array to store the values
  $xmlValArray = array();
  $cnt = 0;
  //loop student element
  foreach ($domXPath->query('//student') as $keyDOM) {
  //loop the attributes name
  //check each attribute has value, if not insert empty string
         for($i=0; $i<count($xmlAttrArray); $i++)
                $xmlValArray[$cnt][$xmlAttrArray[$i]] = $keyDOM-
  >hasAttribute($xmlAttrArray[$i]) ? trim($keyDOM->getAttribute($xmlAttrArray[$i])) :
  '';
         $cnt++;
  }
  //var_dump($xmlValArray);
```

The above examples represent a set of codes to explain some of the functionalities of this mobile application. The complete set will be made available for mobile development on obesity related research under General Public License.

4. Results and Discussion

This paper describes the design and development of the mobile version of CHOIS that can collect data in real-time at the point-of-care and aids the school nurse for obesity surveillance, control and intervention through relevant research on obesity. This application, mCHOIS, has been tested using an Android-based Google phone G1. However, any Android-based phone can be used for this application. A series of system integration and validation were conducted for such testing. All the fields for data input in the form were tested and validated for data type and data format. These formats were same as that of the web based forms of CHOIS [6, 7]. Testing included verifying that the generated XML string follows the standard and be able to communicate with the remote server and the server could read the XML string from the smartphone and parse it. The system also checked for data duplication. In addition, the system validates the data and rejects if a data is not within the right range. For example, the birth date of a school-going student can not be more than 24 years of age or less than 2 years of age. The system also flags when the BMI of an individual is more than certain value. Moreover, certain fields (indicated by *) are essential to be filled-up in the web based form before submitted. Once the data is received by the remote server, it checks for the duplication and then sends the appropriate message to the phone device. If successful, the message reads "..successfully inserted to the database", unless the "..record already exists". Once it is in the database, the web based form of the individual is populated with the real-time data and can be viewed through the web after role-based authorized entry into CHOIS.

Several layers of security have been implemented in this development, which includes a native Screen Lock mechanism that Android provides to deter unauthorized use of the phone. Android platform maintains that a user can not access another user's private data, unless the user explicitly declares permission for others to interfere. After installation of mCHOIS, the user will need to activate the application by entering his/her username and password that is given by the IDHS administrator via email. The application will then create a connection, using SOAP as described in ksoap2-android package, with the server to authenticate the user. If authentication is successful, then this information will be stored in the SQLite database so that the user can use the application to gather some patient data even when there is no network coverage. These data will be stored locally on the phone database. When it is time to send those data, the application will first send user's authentication information to the remote server to verify that the user's authentication is still valid. Then mCHOIS application will gather the data into a single package in XML-format, encrypt the package using Data Encryption Standard (DES) provided by javax.crypto package, and send it to the server. Both Android-client and server side will agree on an encryption method and share the encryption key so that they can exchange information. Presently, mCHOIS has been designed not to transmit any patient-sensitive data. Moreover, the identity of the individual is replaced by a 10-character student ID. Nevertheless, for future enhancement, we are evaluating a third party software, such as, *Mocana* for encryption and/or mobile defense for an extra security and privacy protection.

For its School Health Program, the IDHS provides comprehensive primary healthcare for medical, dental, and mental health to the students through its 39 school-based health centers (SHCs) located throughout the state. Although SHCs allow students to be evaluated, diagnosed, and treated on-site, some of the healthcare related activities including health education for intervention are run off-site where lack of internet connection becomes a limiting factor for digitally collecting the data and evaluation of an individual's health condition including BMI in real-time. mCHOIS, which is now under deployment for the School Health Program in Illinois, will provide the ability to capture data in the field for simultaneously assessing behavioral, physiological, and psychological states of an obese patient in real-time.

Emerging software platform offers the potential that can blur the line between web service and native application [19]. Among the emergent technologies for mobile application, HTML5 [20] was evaluated while developing mCHOIS. The HTML5 technology, which stems from the WHATWG (Web Hypertext Application Technology Working Group), offers to bring more flexibility of Web 2.0 and enhance a wide range of features in a web page. Adding images and video to a web page and make it mobile compatible will be easier, irrespective of the phone's operating system. More importantly, HTML5 would allow storing offline information locally in the mobile phone. However, at present, lack of support by the browsers (IE, Chrome, Firefox, etc.) augmented by the lack of fully matured specification for HTML5 has made the developers adopting this technology difficult. Moreover, while HTML5 local storage feature may provide a cross-platform solution, there is a chance that an HTML-based solution would compromise on speed, usability and reliability. On the other hand, the choice of Android-based platform for this implementation should allow for widespread adoption of this mobile application. Recent announcement of

Microsoft to support HTML5 in the future version of IE[d] may change this strategy. Further development of mCHOIS will adopt HTML5 technology once it is matured.

5. Conclusion

We are developing mobile application for CHOIS using both Android-based smartphone and tablet where appropriate. In our experience, a smart phone can effectively be used for a low-volume data, whereas a tablet-based application can be designed for high volume, complex data entry, such as, for risk-analysis of adolescents. By democratizing health data management and widening its availability, mobile technology has the potential to revolutionize telemedicine, clinical practice, medical education and information distribution, particularly in rural areas, and to make patient-centric medical computing a reality. mCHOIS, which is now under deployment in Illinois for IDHS, will be further enhanced by utilizing the various features of Android-based phone for supporting obesity intervention program. With the built-in GPS capability for location based services, the at-risk individuals will be able to better control their weight by locating the access to the healthy foods, as for example. Moreover, this GPS capability along with its built-in camera will be useful for point-of-care, particularly in emergency.

Acknowledgements

A part of this work was submitted by Andi Sumargo, and Dheeraj Petla for MS (Computer Science) Capstone Project. Authors are thankful to Weimo Zhu, Radha Nandkumar, Ogun Tigli, Sumit Vyas, Sameera Palnati and Mudigonda Manjunath for helpful discussions. Special thanks to Yumiko Iwai and Jocelyn LaRouche for suggestions during the development of this application. The authors also appreciate the support of Dr. Thomas MacCalla.

References

1. R. Gardiner, Stud Health Technol Inform, 137 (2008) 241-56.
2. I. Kouris, S. Mougiakakou, L. Scarnato, D. Iliopoulou, P. Diem, A. Vazeou, D. Koutsouris, Int J Electron Healthc, 5 (2010) 386-402.
3. M.Y. Mann, M.A. Lloyd-Puryear and D. Linzer, Pediatrics, 117 (2006) S315-9.
4. B. Holtz and P. Whitten, Telemed J E Health, 15 (2009) 907-9.
5. E. Arsand, J.T. Tufano, J.D. Ralston, P. Hjortdahl, J. Telemed Telecare, 14 (2008) 329-32.
6. A.K. Datta, V. Jackson, R. Nandkumar, J. Sproat, W. Zhu, and H. Krahling, In T. Solomonides, I. Blanquer, V. Breton, T. Glatard, and Y. Legre (eds.), Healthgrid Applications and Core Technologies, IOS Press, Netherlands, 2010.
7. A.K. Datta, V. Jackson, R. Nandkumar, W. Zhu, Proc of the PRAGMA 18, San Diego, 2010.
8. S.Z. Yanovski and J.A. Yanovski, New England Journal of Medicine, 346 (2002) 591-602.
9. Y. Wang, M.A. Beydoun, L. Liang, B. Caballero, S.K. Kumanyika, Obesity, 16 (2008) 2323-30.
10. S.E. Barlow and W.H. Dietz, Pediatrics, 102 (1998) e29.
11. L.A. Campfield, F.J. Smith, P. Burn, Science, 280 (1998) 1383-7.
12. A.J. Nihiser, S.M. Lee, H. Wechsler, M. McKenna, E. Odom, C. Reinold, et. al., J Sch Health, 77 (2007) 651-71.
13. C. Haseman (ed.), Apress, Android Essentials (Books for Professionals by Professionals), 2008.
14. J. Stark (ed.), O'Reilly Media, Building Android Apps with HTML, CSS, and JavaScript, 2010.
15. M. Dalrymple and S. Knaster (eds.), Learn Objective-C on the Mac (Learn Series), Apress, 2009.
16. Hong Zhu (ed.), Software Design Methodology: From Principles to Architectural Styles, Butterworth-Heinemann, Massachusetts, 2005.
17. Reza B'Far (ed.), Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML, Cambridge University Press, Cambridge, 2004.
18. Deborah J. Mayhew (ed.), Principles and Guidelines in Software User Interface Design, Prentice Hall, New Jersy, 2008.
19. L. Constantinescu, R. Pradana, J. Kim, P. Gong, M. Fulham, D. Feng, Conf Proc IEEE Eng Med Biol Soc., 2009 5167-70.
20. M. Pilgrim (ed.), HTML5: Up and Running, O'Reilly Media, 2010.

[d] http://ie.microsoft.com/testdrive/