Scripting languages and technologies for mobile application development

SERENA PASTORE
INAF -Astronomical Observatory of Padova
Vicolo Osservatorio 5 – 35122 Padova)
ITALY

Abstract: - When using standard web technologies to develop web applications, a big constraint is the context in which the application should run. In particular the mobile context is very different from the desktop one even in the design of web applications. Many aspects should be taken into consideration related to hardware and software constraints to which every mobile device is subjected. The paper describes the choice of adopting standard web technologies in developing an application for the mobile framework and specifically the decision to use a standard scripting language such as ECMAscript for developing an application running within a browser environment devoted to mobile devices. This is the result of the analysis of issues and problems related to development in the mobile context.

Key-Words: - Web standard technologies, scripting languages, mobile web, web design, web programming languages

1 Introduction

Mobile devices are becoming more and more popular and widespread, especially among young people. Reaching this category of users is essential for our group, which for many years has used emerging Internet technologies for outreach and education about Astrophysics [1]. To celebrate the “Year of Astronomy” (http://www.astronomy2009.org), which will start next year, many initiatives are in the development phase. In particular we are working on a project about the realization of an interactive course called “The Virtual Planetarium” to be deployed on mobile devices. This application is designed as a “quiz” with a set of questions about our solar system and its planets throughout explanations of the wrong and the right answers. To deploy, such an application on mobile devices, many issues about specific constraints both on the hardware and software aspects of this kind of device arise. Development in a mobile environment [2] involves several choices for the content format and the methods used for its development, which include a different platform, techniques and programming languages heavily based on mobile devices types and their features and support capabilities. The mobile computing world is very different from the desktop or server computing world not only due to hardware characteristics, but also due to the lack of homogeneity in software features, which could be outlined in the general computing world.

Mobile devices are very dependent on their manufactureres, which impose not only commercial operating systems but also various levels of tools to build software applications devoted to their terminals. Attempts to create a standard software platform (at least three or four macro areas as in desktop software) have met without success. Three categories should be taken into account when developing an application for a mobile environment:

- Broad categories of micro-devices, each one with different functionalities and capabilities: players (mp3/audio-video), cellular phones, PDAs, smartphones, low-end notebooks as a means of reduced features such as video screens or CPU/memory equipment.
- Network connections (wireless, cellular or satellite connections) [3]
- Software environment in terms of the operating system and application software (players, browsers, etc.) available.

The paper analyzes the main issues and problems related to development in the mobile context and describes the solution adopted in the re-design of the “Virtual Planetarium” applications devoted to such type of devices.

2 The mobile environment

The mobile computing environment presents many aspects that make it a particular framework in which to develop applications. Mobile devices have reached a great diffusion, but the applications
deployed on these devices have not followed the same trend due to hardware and software constraints that limit this category of software. The mobile environment is very closed by manufacturers that impose technology and limit the development of open source software running on such devices. Thus it is very difficult and sometimes even impossible to develop an application able to run within every kind of mobile device both for hardware and software issues.

Focusing on software aspects, the difference between a standalone application and a web application that may use server-side technology depends particularly on the type of network connection. A network connection on such a device could be of a different kind, intermittent or not intermittent and however yet too expensive for most users. We first analyze the issues from a software perspective.

2.1 Mobile operating systems
There are a range of operating systems running on mobile devices. Some are more closed to desktop operating systems such as the Microsoft version of its operating system, a.k.a. Windows Mobile, or the project trying to port Linux-like operating systems on mobile devices (i.e., the Moblin project, http://www.moblin.org, focused on building a Linux-based platform optimized for mobile devices or the Ubuntu Mobile Internet Device Edition, http://www.ubuntu.com/products/mobile). Others rely on manufacturers like Symbian (http://symbian.com) and PalmOS (http://www.palm.com), which are operating systems strictly related to the devices’ manufacturers. The first category of operating systems poses fewer constraints on development, since it is possible to have a framework next to the desktop one: unfortunately, a great many mobile devices used by most people (i.e., Nokia devices) rely on the other category.

2.2 Mobile developing frameworks
Development in the mobile environment should take into account the different technologies and the different kinds of tools, some of which are related to specific platforms. We could distinguish manufacturers service development kits (SDKs), programming languages frameworks and specific languages used for applications based on web browsers. Table 1 shows a comparison between the different approaches, outlining the main requirements and limitations.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Requirement</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Lite</td>
<td>Player</td>
<td>Not free for all devices</td>
</tr>
<tr>
<td>J2ME</td>
<td>JVM machine</td>
<td>Difficult development (requiring skills)</td>
</tr>
<tr>
<td>BREW</td>
<td>Development environment</td>
<td>Commercial platform that does not cover all devices</td>
</tr>
<tr>
<td>Web technologies</td>
<td>Browser</td>
<td>Technologies supported by mobile browsers</td>
</tr>
</tbody>
</table>

2.2.1 Service development kits
Mobile devices’ manufacturers provide specific development platforms (SDKs) related to the manufacturers’ operating systems to programmers in order to develop applications for the manufacturers devices. The issue is that the portability of such applications is limited or impossible.

2.2.2 Programming languages frameworks
Programming languages refer to common languages used for development in the desktop environment. According to their nature (compiled or interpreted languages), they require a framework in which to deploy and execute the applications. Some languages have a specific version devoted to mobile devices that consist of a “reduced:” version of the main language in term of available libraries.

An example is the Java platform devoted to mobile devices called J2ME (Java 2 Mobile Edition) [4] which provides specific libraries for this kind of device and a reduced virtual machine (JVM) on which to execute and run programs. Another example is the Adobe Flash Platform, which with the Flash Lite edition [5], provides an environment in which Flash applications can work. The problem is that while a Java bytecode is a portable program executable on any JVM regardless of the platform on which the bytecode was originally compiled, Flash applications require a player that is not freely available and/or installable on mobile devices platforms. Finally, the Binary Runtime Environment for Wireless (BREW) is an application development platform created by Qualcomm for mobile devices (http://brew.qualcomm.com). It is a set of APIs that enables software development in C/C++ languages giving control over the handset hardware: however BREW is a commercial platform. Such a solution
provides a customized application for a limited number of devices.

2.2.3 Browser-based and web-based applications: markup and scripting languages

A special category refers to applications specifically developed to run on a web browser generally called web applications since they are executed through the web and thus follow the HTTP protocol. Most applications, even if they are devoted to the desktop environment, present a web interface in order to be executed on a web browser. Many of the “so-called” web applications, however, do not require a network connection (i.e., those developed with only client-side technology, which requires only a browser locally installed on the device), but they fall into the same category. In this way, web standards have become a great technology to be used for application development. This affects three different aspects: content structure, presentation and behavior. Technologies involved in such development are XHTML (data content), CSS (the view) [6] and several scripting languages, both client and server side, used to develop such applications (the behavior).

Here we can distinguish specific languages devoted to structure and visualize content interpreted by the browser (the markup languages) from those leaning toward the logic in which there are a lot of programming languages, both client and server side, also derived from those used for the desktop environment.

Markup languages

In the first category of languages, the first implementation of a language devoted to mobile devices was the WML (Wireless Markup Language) [7] developed by the WAP Forum inside the OMA, the Open Mobile Alliance works (http://www.openmobilealliance.org), an organization of industries that works on every aspect of the mobile environment by producing open standards. Such a language was developed to work within the Wireless Application Protocol (WAP) [7], the protocol by which mobile devices communicate. WML applications are stored on an ordinary web server such as the XHTML-based applications that use HTTP protocol, but WML applications are intercepted by a WAP gateway that sits between mobile devices and the web passing pages like a proxy. This process is hidden from the device, and the application is accessed in the same way by the browser through its URL.

Many developers tend to use XHTML as the markup language with the WAP protocol. XHTML is the standard markup language (as the XML reformulation of HTML) defined as a specification by the W3C organization (http://www.w3c.org) used for structure content on the web through Ipermedia. Inside the W3C, the Mobile Web Initiative (MWI, http://www.w3.org/Mobile) collects the main effort about content production and access from mobile devices and the creation of mobile-friendly applications by developing best practices based on web standards. Specifically, the “Mobile Web Best Practices” (MWBP) (http://www.w3.org/TR/mobile-bp/) describes how to develop content accessed from a mobile device which becomes a W3C recommendation, and promotes the use of XHTML to structure content for mobile devices. Indeed, the W3C’s and the OMA’s work is connected since the two organizations collaborate and converge in the new version of WML (2.0), also called XHTML-MP (Mobile Profile), which has been developed as a subset of the basic XHTML tags. In the Web/XHTML solution (the web used as a content delivery tool), content is modified to suit the characteristics of micro devices, and application execution requires only a web browser.

Scripting Languages

Typically, a platform-independent scripting language, as shown in Fig.1, uses a runtime player to communicate with a computer’s operating system and graphical user interface (GUI). Scripting means the orchestration of events, action, and behavior of objects.

Fig. 1: Software environment for scripting languages.

In terms of logic, many technologies can be used to develop applications for both the client and server sides that are related to the fact that logic is ported on the server and thus executed through the web server and its modules are able to run applications or on the client, and thus are executed within the browser environment.

Server-side technology provides many languages, most collected under the concept of scripting languages by means of a set of instructions whose runtimes are interpreted by a specific interpreter.
There are numerous scripting languages, also called dynamic languages because they are interpreted and compiled at runtime. Most follow the object-oriented programming paradigm, and the most prominent are the PHP, Python and Ruby programming languages [8]. Developers overseeing these languages continue to map out improvements. All require the presence of the interpreter and the enabling of related modules on the web server.

From the client side, a specific programming language that is now a standard is ECMAScript (http://www.ecmascript.org/) with its most known implementation, which is Javascript. These scripting technologies, which are interpreted by most engines’ browsers, could be used to develop an application able to run on every device presenting a web browser.

Table 2 shows the most used browsers (not proprietary, which could be found or installed in mobile devices). All support client-side scripting technology.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Explorer</td>
<td>Windows-like</td>
</tr>
<tr>
<td>Mozilla Minimo</td>
<td>Windows/Linux</td>
</tr>
<tr>
<td>Opera Mini</td>
<td>Windows/Linux</td>
</tr>
</tbody>
</table>

3 Choosing a client-side scripting language for application development

Scripting languages are general-purpose programming languages most used for web application development. They differ from the compiled programming languages since instructions are executed in “real-time” by an interpreter, reducing the developing phases and with the possibility of easy porting of the software on different hardware and software platforms.

In order to have an application that could reach the largest number of devices, the choice of a client-side scripting technology could be considered the best solution. An application designed based on this technology does not necessarily require a constant network connection (only if in the first phase of downloading), such an application requires a browser able to interpret an implementation of ECMAScript, and most web browsers found in mobile devices are able to do so. In this way, such a solution could be flexible and portable, and thus not related to a proprietary solution. This is an important aspect since the initial idea to develop the application as a J2ME application or Flash application is strongly related to the possibility of installing and running it. For example, Nokia devices limit the installation of Java applications on their devices and thus could be a problem. A Flash application indeed is related to the presence of the specific player. By using a web-browser based solution based on the web standard, attention is devoted to make the content as much as possible closed to web standards, which could make the content independent from hardware capabilities in terms of screen dimension, processing and memory capabilities.

3.1 The proposed solution

The application concerns an astronomical course developed as a set of about 40 questions on the main arguments. Users follow a specific path answering such questions, each one presenting a specific explanation about the wrong and right solutions with some deeper explanations. Structured on different levels that correspond to a specific argument, the application includes the option of saving a specific level, and thus users can start from a specific level the next time they decide to continue the course.

The web technologies seem to be the best choice to develop applications devoted to mobile platforms and could be executed on most devices since these technologies provide a web browser. However the use of server-side technology is related to a network connection which could be very expensive. A tradeoff could be client-side programming that allows for executing the application on browsers. With this option the application has been tuned to mobile devices by using web standards technologies to structure and present data and to provide the application logic [9].

Content is structured and presented through XHTML and CSS technologies in order to be visualized through a web browser (Fig. 2). Since the application is exclusive to mobile platforms, some tricks should be adopted to provide a good visualization: for example, the images’ size and format (which should be reduced), specific attention paid to scrolling and page navigation, avoiding the use of specific tags and HTML elements such as tables for layout, frames or image maps, etc.

The elaboration of answers should be done with a specific programming language such as Javascript adopting a remote scripting through the model of XHTML documents given by the Document Object Model (DOM) as specified by the W3C (http://www.w3.org/DOM/).
This model provides an interface that will allow programs to dynamically access the document since it could be represented as a hierarchical tree. The choice of using only standards could guarantee the portability of such an application on different hardware and software platforms.

### 3.2 ECMAScript and Javascript

The ECMAScript language is the standardized specification (ECMA-262) of the ECMA (European Manufacturers Association http://www.ecma-international.org) of the major dialects referred to as Javascript or Jscript [10]. The language is supported in many applications, especially web browsers, and is commonly known as Javascript. The dialects (Jscript by Microsoft, ActionScript by Adobe, etc) differ by standard libraries: some are standardized separately such as the W3C-specified DOM, but they are designed to be compatible and standard-based. As the industry-standard, must-know scripting language, Javascript is supported by all major browsers and is increasingly the foundation of new web development technologies. Its use has in fact increased, after a bad reputation since problems about an improper use when using the opening windows phenomena (i.e. popus) or for security reasons, since its development and inclusion of new improvements (i.e., the XMLHttpRequest object) has led to its application in the so-called Ajax-based applications, which are the basis of rich interactive applications that create a rich user experience with the agility of desktop applications. Javascript and its object are used in Google-based applications and thus have contributed to its use. An example of an scripting program included in the web page as an external file (in order to guarantee a separation between content, presentation and logic) is shown in Fig. 3

```javascript
<script type="text/javascript">
function check_input() {
    var odd = document.getElementById('odd');
    if (odd.value == document.getElementById('even')) {
        window.location = "domanda/odd.html"
    } else if (odd.value == "domanda/even") {
        window.location = "domanda/od.html"
    } else if (odd.value == "domanda/ed.html") {
        window.location = "domanda/ed.html"
    } else if (odd.value == "domanda/ed.html") {
        window.location = "domanda/ed.html"
    } else if (odd.value == "domanda/ed.html") {
        window.location = "domanda/ed.html"
    } else {
        alert("Insieme il numero e il relativo contraffermo.")
    }
}

function testo(event) {
    if (event.keyCode == 40) {
        window.location = "inizio.html"
    } else if (event.keyCode == 39) {
        window.location = "accordo.html"
    } else if (event.keyCode == 37) {
        window.location = "inizio.html"
    } else if (event.keyCode == 38) {
        window.location = "accordo.html"
    } else if (event.keyCode == 39) {
        window.location = "inizio.html"
    }}
</script>
```

Fig. 3: Javascript code example implementing the question (developed with Javascript instruction using the DOM model for the XHTML document)

Each question is implemented by using the DOM model and Javascript-related methods and properties to access the HTML form’s elements, which represent a different solution to the questions.

### 4 Conclusion

Choosing the right framework to develop applications for a particular environment such as the mobile devices could be difficult. The choice affects software architectures suited to application development and software development methodology, and includes the maturity of the software and the community, and the richness and completeness of documentation, including samples and user support. An additional issue involves the variety of existing platforms, which makes it very difficult to choose a platform on which to develop applications that are portable on different devices. Most of the available software platforms are tied to a specific device and thus reduce their use. The use of a web application could in part solve the problem of porting on different technologies.

Further debate about scripting languages being better than proper languages continues. Scripting languages are used in some scenarios, but may not serve all enterprise software requirements. The choice of technology depends on the answers to key questions: the effectiveness in solving the problem at hand by speeding up the process by providing off-the-shelf components as building blocks to form the
solution; if it proves itself in a demanding production environment addressing non-functional requirements of performance and stability. It is important not to get locked in with the vendor, but to consider the features a particular script offers before deciding on whether or not to use it. Some languages have been designed with features that will keep the languages more current with the latest technologies. Scripting languages can be enhanced with add-on components. Programming scripts are a very important element of web designing: they are powerful tools and can be used to enhance the functionality of the site. They determine the level of interactivity the visitors will have. The choice of implementing an application with a scripting language can lead to problems regarding behavior on the part of management, but then could be successful if software engineering practice is strong. Choosing the solution of a web application developed with a client-side scripting language could reach the largest number of devices.

References:


[7] Ronan Mandel and Paul Fonte, WAP Development with WML and WMLscript, Sams, 2000

