THE WEB AS A COMMON LANGUAGE TOWARDS SOFTWARE SUSTAINABILITY

Andreas Kamilaris
Networks Research Laboratory, University of Cyprus
Department of Computer Science, Nicosia, Cyprus

Andreas Pitsillides
Networks Research Laboratory, University of Cyprus
Department of Computer Science, Nicosia, Cyprus

ABSTRACT

In this paper, we explore the potential of the Web to constitute a common language in various computing domains towards increasing software sustainability. We discuss various benefits of Web development, which offer sustainable characteristics to software. We demonstrate two case studies, in which the Web can be used for software development. The first is a mobile application offering tourist guide services and the second is a real-world pervasive framework, targeting home automation in a smart home environment. Our experimental results show that both software operate in satisfactory performance, showing that the benefits of Web development can be acquired without significant compromises in response times.

KEYWORDS

Web, Software Sustainability, Mobile Web Development, Tourist Guide, Web of Things, Smart Home.

1. INTRODUCTION

Generally, enterprises around the world tend to develop their own, proprietary software for providing specialized solutions to individuals and other companies. In this way, they manage to keep their customers engaged and dependent on their platforms and systems, ensuring high profits in the long-term. This creates the phenomenon of *vendor lock-in*, making a customer dependent on a vendor for products/services, unable to use another vendor without substantial switching costs. Software updates and adaptation to more modern technologies means added costs to the customer in order to keep pace with technological advancements. This means that true software sustainability is not in the interest of most companies.

Sustainability in general refers to ensuring that a development meets the needs of the present without compromising the ability of future generations to meet their own needs. In this context, software sustainability can be defined as the ability of a designed product to operate over long periods of time without degrading itself or the environment (Markiewicz, 2012). To accomplish this goal, sustainable products emphasise features like efficiency, durability, reusability and modular construction.

In the early 1990s, public, royalty-free standards were proposed as a solution to vendor lock-in, in an effort to increase software sustainability mainly through durability and reusability. However, software vendors used "embrace, extend and extinguish" tactics to achieve a dominant market share or to monopolize a product category, by entering product categories of popular standards, extending them with proprietary capabilities, and then using those differences to disadvantage their competitors, rendering the standard obsolete in some cases, e.g. the Microsoft case (The Economist, 2000).

Since this solution did not work, the use of free, open-source software has arisen in the 1990s as a stronger alternative. Because open-source can be modified and distributed by anyone, the availability of code cannot tie a user to particular distributors. This practice makes vendor lock-in less effective, since there is no incentive for developers to invent redundant new formats and protocols if usable, royalty-free standards exist.

Although open-source code enhanced even more the efficiency, durability and reusability of software, there is still a long way to go to achieve software sustainability.

In latest years, the Internet has penetrated deeply in our lives while the Web is used extensively by people for work or amusement. The Web is ubiquitous, scales particularly well and has well-accepted and understood tools. The popularity of Web development ensures reusability of software, the wide variety of well-established libraries ensures software durability, while the advancement of Web browsers and languages promises good performance and code efficiency. Hence, we suggest that the Web can become the common language in order to achieve true software sustainability, and that it can be applied in various software development domains, such as mobile computing and real-world applications.

Critics of the potential generality and applicability of the Web in other domains argue that its performance becomes limited in any use other than classical Web browsing. In this paper, we examine whether the Web can be successfully applied in two modern , popular areas: mobile and pervasive computing, in order to provide sustainable characteristics to software in these areas too.

The rest of the paper is organized as follows: Section 2 discusses the general reasoning for using the Web as a common language towards software sustainability while Section 3 presents two real-world applications, one for mobile computing and one for home automation, which have been developed by using Web technologies. Finally, Section 4 defines future work and concludes the paper.

2. USING THE WEB TOWARDS SOFTWARE SUSTAINABILITY

The great success of the Web as a global platform for information sharing educates that simple and loosely-coupled approaches offer a high degree of scalability, robustness and evolvability. This success is partly due to an open and uniform interface, which enables even non-experts to develop interactive applications rapidly. Unlike most protocols used in specialized domains such as embedded computing, Web standards are widely used on the Internet, being open, well-understood and highly flexible.

Using the Web as a common language in various computing domains offers several benefits, such as wide-spread deployment of Web browsers and ubiquitous HTTP support in programming and scripting languages. Users can employ any computing device to manage their environment (e.g. mobile phones, tablets, smart TVs, PCs, home routers, smart meters), provided that these devices have access to the Internet/Web.

Moreover, open access to systems and applications through Web services (WS-*, REST), using open and simple standards and data formats (e.g. XML, HTTP, JSON) enables information to be reused across independent systems, lowering the access barrier that allows people to develop their own composite applications. People may combine more complex services following the classical technique of Web mashups.

3. CASE STUDIES OF WEB DEVELOPMENT IN OTHER DOMAINS

In this section, we demonstrate two real-world applications, targeting computing areas other than the classical Web browsing, where Web technologies can be applied with success, offering various benefits. The first is a mobile application about a tourism guide and the second is a framework for home automation using embedded sensors and actuators. By means of these developments, we aim to show how Web programming can be employed to achieve the required tasks, offering high interoperability, flexible design, reduced costs due to open-source code available and ease of implementation, ensuring sustainable software development with increased life cycle of the product, because of the current popularity of the Web, its promising future and its stable, scalable and well-established structure.

3.1 A Tourism Guide Mobile Application

Tourism is an important source of income for many countries, especially those combining culture and relaxation. Cyprus, a small island in the Mediterranean, is one such example, counting on tourism for its financial growth. Cyprus blends archaeological wealth with beautiful beaches and warm temperatures and it is a popular destination for tourists globally. In order to offer better services to tourists, in regard to informing them about the archaeological and religious attractions of the island and its beautiful beaches, we developed a tourism guide mobile application (Networks Research Laboratory, 2013). This constitutes a sustainable environmental action, since no paper is needed for printing books or magazines serving as guides.

We implemented the mobile application by using Web technologies, namely HTML5 and AJAX. By means of Adobe PhoneGap[1], which is a free and open-source framework that allows to create mobile applications in multiple phone platforms, we created the tourist guide application being able to operate on Android, iOS (for iPhone, iPad), Windows, Nokia phones (Symbian-based), Blackberry and WebOS. PhoneGap helps to acquire the current location of the user, using Wi-Fi, GSM or GPS localization.

By employing the Kendo UI framework[2], which is a comprehensive HTML5/JavaScript framework for modern Web and mobile application development, we managed to select from various designs, themes and layouts which look like a native application on the targeted platform. Figure 1 shows two different versions of the mobile application, one for Android and one for iOS. The reader can observe how similar both versions look to a native application on these platforms. By going native, we would be obliged to learn specific programming languages and methods, such as Java for the Android case and Objective-C for iOS.

We list below the main features, all enabled by means of modern Web techniques and tools:

- View nearby attractions from your current location, either on a list or on the map of Cyprus.
- See detailed descriptions of particular attractions by clicking on them, as well as relevant information such as opening hours, contact details, entrance fees etc.
- Filter attractions according to the user's preferences, to view only places of interest.
- Adjust distance from nearby attractions. It can range from hundreds of meters to tens of kilometers.
- Search for attractions on a lexicographical basis.
- View the most popular attractions in every city or around the island, as they are rated by other tourists.
- Create a tour plan, by adding attractions to a list. This list can be edited even before entering Cyprus, to better organize vacations.
- Mark attractions as visited, to keep a list of visited places as a reference.
- Rate a visited attraction, so that other tourists would be assisted to select only the best places.
- View directions to an attraction, either on the map of Cyprus or in the form of text directions.

We note that the tourist's profile (sex, age, nationality, preferences), as well as his visited attractions, are saved anonymously on a Web server, by means of HTTP calls. Thus, our mobile application offers more personalized suggestions to tourists, such as places visited by people of similar age or same nationality.

Concerning some figures of use, in two months of our mobile application being on Google Play (Networks Research Laboratory, 2013), it has been used by 1,500 tourists from 30 different countries. This indicates some success, especially since most tourists being happy about it, have rated it positively (rating 4.0/5.0).

Comparing the performance of our mobile application with a very similar native one (Quinto Stdio Inc., 2013), offering very similar services, our application is slightly slower, but only in a small degree (10-16%). All actions and features are performed in a few seconds, while the menus are loaded in milliseconds.

3.2 A Smart Home System

Embedded computing is largely characterized by vendor-specific standards and specifications, since each manufacturer delivers his own solutions. However, recent efforts for porting the IP stack on embedded devices (Hui, 2008) and the introduction of IPv6, which provides extremely large addressing capabilities, can facilitate the merging of the physical and the digital world, through the Internet. Inspired by the potential Internet-enabling of embedded devices, sensors and actuators, the Web of Things (WoT) (Wilde, 2007), (Guinard, 2010) reuses well-defined Web techniques to interconnect this new generation of physical devices, following the REST architectural protocol (Richardson, 2007).

To demonstrate the operation of a smart home using Web technologies, based on the WoT, we developed a lightweight, Web-oriented application framework providing uniform access to heterogeneous embedded devices via standard HTTP calls (Kamilaris, 2013), (Kamilaris, 2011). The framework supports multiple Web clients and numerous smart devices, placed inside the home environment. Detailed information about the operation and implementation of the framework are available in the technical report in (Kamilaris, 2012).

1 Adobe PhoneGap. http://phonegap.com/
2 Kendo UI Framework. http://www.kendoui.com/

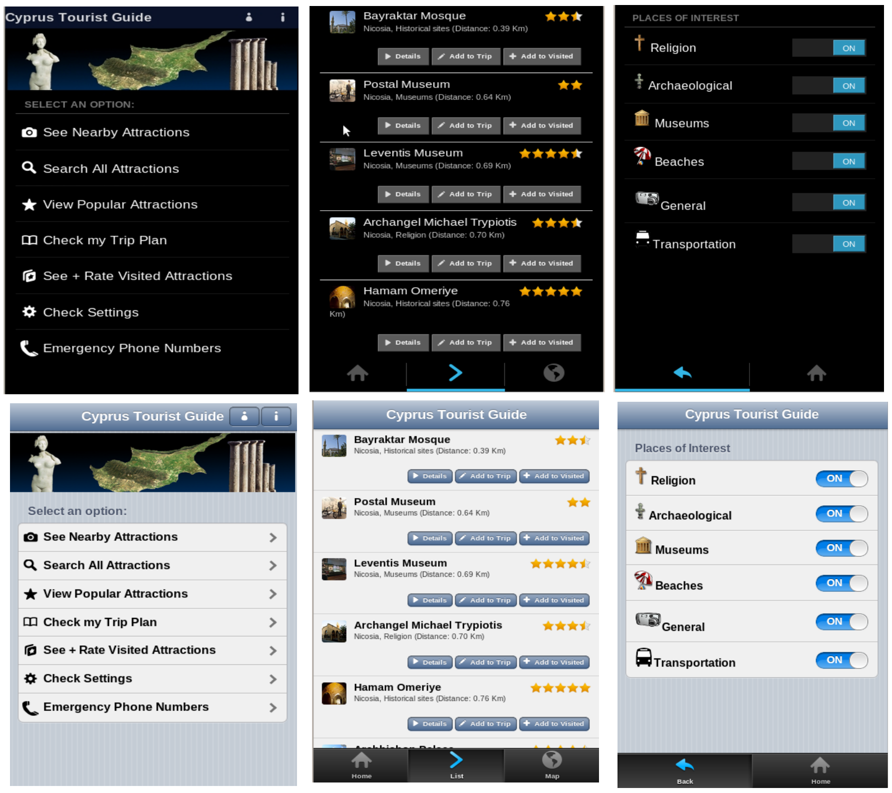

Figure 1. Snapshots of the tourist guide mobile application. The Android version appears on top, while the iOS version on the bottom. From the left to the right, the application shows the main menu, a list of nearby attractions and a filtering of attractions according to personal preferences.

Figure 2 (left) depicts the general architecture of this framework. It follows a layered model and it is composed of three principal layers: *Device Layer*, which is responsible for the management and control of embedded devices, *Control Layer*, which manages the whole system and *Presentation Layer*, which generates dynamically a representation of the available devices and their services to the Web, enabling the uniform interaction with them over a RESTful interface. A *Web Server* allows real-time interaction while a *REST Engine*, implemented by Restlet[3], ensures a RESTful environment. Thanks to REST, Web clients can easily explore available devices and their services by clicking links, as they would browse Web pages.

The embedded devices used in our smart home system are Telosb sensor motes[4], for sensing environmental conditions (e.g. temperature, humidity, illumination), and Ploggs[5], which are smart power outlets for controlling electrical appliances (by switching them ON/OFF) and sensing their electrical consumption. Telosb motes have been programmed using TinyOS 2.x[6], which is an operating system for low-power embedded devices. By means of Blip[7], which is an implementation of the 6LoWPAN stack for TinyOS, we exposed the sensing capabilities of the motes as RESTful Web services, transforming them into embedded Web servers. 6LoWPAN (Kushalnagar, 2007) is an adaption layer that allows efficient IPv6 communication over the IEEE 802.15.4 standard. Through 6LoWPAN, an IPv6-enabled wireless sensor network was formed.

Ploggs are smart power outlets that can sense the consumption of electrical appliances in real-time. A Plogg can be attached to any electrical appliance or device that uses a standard mains socket plug. Through a ZigBee chip, Ploggs can also create a wireless multi-hop network inside a smart home.

---

3 Restlet framework for REST. http://www.restlet.org/
4 Telosb sensor mote datasheet. www.willow.co.uk/TelosB_Datasheet.pdf
5 Ploggs Smart Power Outlets. http://www.prnewswire.co.uk/news-releases/consumer-products-retail-latest-news/plogg-the-configurable-zigbee-smart-plug-155674645.html
6 TinyOS Operating System for embedded devices. http://www.tinyos.net/
7 The Blip Project. http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip

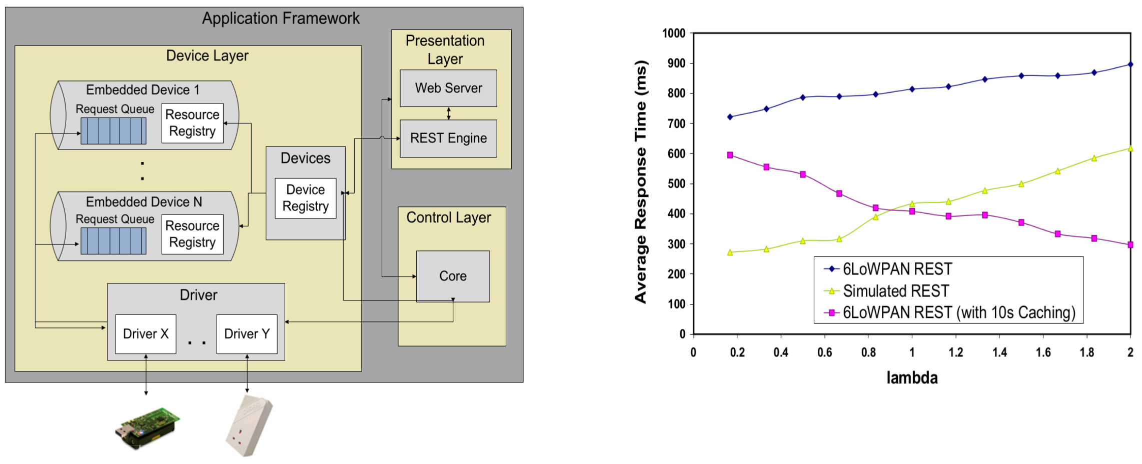

Figure 2: Architecture of the smart home application framework (left). Response time in Web vs native case (right).

Figure 2 (right) shows the response times in varying number of arrival requests per residents (lambda = requests/second), in the 6LoWPAN case (6LoWPAN REST) in comparison to a native implementation (Simulated REST). These tests focus exclusively on the wireless network of Telosb sensor motes. Although the native case is almost 2x faster in some tests, in both implementations response times are less than a second, even in high traffic. Hence, the Web-based case is feasible with acceptable performance. When HTTP caching is employed (which is a basic feature of the Web), with 10 seconds freshness time, 6LoWPAN REST executes faster than simulated REST, when lambda>0.8 requests/second. More information about this experiment is available in (Kamilaris, 2013).

On top of the aforementioned infrastructure (application framework, Telosb sensor motes and Ploggs), we developed the following applications, using Web technologies:

- A Web interface allowing residents to view their electricity footprint in real time (Figure 3, left).
- A graphical editor for creating smart rules for home automation (Figure 3, right).
- A task scheduling mechanism adapted to demand response from a smart grid (Kamilaris, 2011a).
- Sharing of home devices and services between online friends through Facebook (Kamilaris, 2011b).

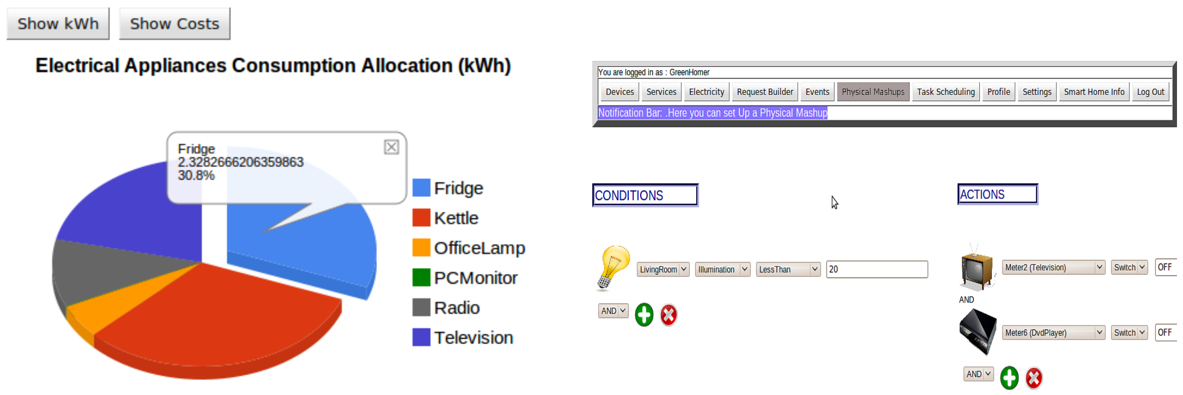

Figure 3: Snapshots of the Web applications built on top of the smart home Web-based infrastructure.

4. CONCLUSION

In this paper, we discussed the potential of Web in constituting a common language in various computing domains towards increasing software sustainability. We demonstrated two such domains, in which the Web could be effectively used for software development: a mobile application offering tourist guide services and a real-world pervasive application, targeting home automation in a smart home environment. In both cases, satisfactory performance in terms of response times is achieved, showing that the benefits of Web development can be acquired without significant technical compromises. For future work, we plan to evaluate in a larger extend the pros and cons of Web development in the aforementioned domains, measure the impact of the Web as a factor for software sustainability and finally, try to demonstrate the use of Web technologies in other computing domains, such as smart transportation, urban computing etc.

ACKNOWLEDGEMENT

The Tourist Guide mobile application is supported and funded by the Research Promotion Foundation of Cyprus, under Grant No. BUSINESS/PRODUCT/0609/73, “Development and Application of a Digital-Historical-Cultural Tourist Guide”.

REFERENCES

Books

Richardson L. and Ruby S., 2007. *RESTful Web Services*. O’Reilly, Sebastopol, Canada.

Journals

Kamilaris et al., 2011. *The Smart Home meets the Web of Things*. International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Special issue on The Smart Digital Home, vol. 7, no. 3, pp. 145-154.

Kamilaris et al., 2013. *Building Energy-aware Smart Homes using Web Technologies*. Journal of Ambient Intelligence and Smart Environments (JAISE), vol. 5, no. 2, pp. 161-186.

Conference papers or contributed volume

Networks Research Laboratory, 2013. *Cyprus Tourist Guide*, Online at: https://play.google.com/store/apps/details?id=com.kami

Guinard et al., 2010. *Architecting a Mashable Open World Wide Web of Things*. Technical Report No. 663, Dept. of Computer Science, ETH Zurich, Zurich, Switzerland.

Hui W. and Culler D. E., 2008. *IP is dead, long live IP for wireless sensor networks*. In Proc. of the Sixth Conference on Networked Embedded Sensor Systems (SenSys), Raleigh, NC, USA, pages 15–28.

Kamilaris A. and Pitsillides A., 2011a. *Exploiting Demand Response in Web-based Energy-aware Smart Homes*. In Proc. of the First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (Energy 2011), Venice, Italy.

Kamilaris A. et al., 2011b. *Lessons Learned from Online Social Networking of Physical Things*. In Proc. of the Sixth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Barcelona, Spain.

Kamilaris A. and Pitsillides A., 2012. *A Restful Architecture for Web-based Smart Homes using Request Queues*. Technical Report No. TR-12-5, Department of Computer Science, University of Cyprus.

Kushalnagar N. et al., 2007. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*. RFC 4919.

Markiewicz P., 2012. *Save the planet through sustainable web design*. .Net Magazine, Online at: http://www.netmagazine.com/features/save-planet-through-sustainable-web-design

Quinto Stdio Inc. 2013. *Tourist Guide for Greece*, Online at: https://play.google.com/store/apps/details?id=Quinto.app.GreekTouristGuide

The Economist, 2000. *Deadly embrace*, Online at: http://www.economist.com/node/298112

Wilde E., 2007. *Putting things to REST*. Technical Report UCB iSchool Report 2007-015, School of Information, UC Berke- ley, USA.