

# A Web Based Approach to Virtual Appliance Creation, Programming and Management

Anasol C Peña Ríos, Jeannette S.Y. Chin, Victor L. Callaghan

University of Essex, UK

E-mail: acpena, jschin, vic [@essex.ac.uk]

**Abstract**— The Internet and Web technology is advancing at a frantic pace, expanding into almost every aspect of our everyday life. One of the latest scientific activities for the Internet and the Web is the so-called pervasive or ubiquitous computing where networking plays a vital role in its core computational framework. In this, people are able to use the Internet and Web to manage the operation of embedded network devices, services and to coordinate their services in ways that create applications such as smart-homes, smart-offices, smart-cars etc, collectively referred to as intelligent environments. For ordinary people (non technologists) to be able to use this technology it is required that the interaction between the users and the environment must be as transparent and simple as possible, employing intuitive and user-friendly interfaces wherever possible. A popular approach to empowering users to customise the functionality of their environments is via end-user programming. In this work-in-progress paper we describe an approach based on using a web based GUI to augment earlier work of ours concerning an end user programming paradigm known as Pervasive Interactive Programming (PiP), in a way that makes it more flexible and easy to use. In doing this we present a conceptual model and discuss the issues in developing and using this model.

*Pervasive Interactive Programming, End-user Programming, Rich Internet Application, Web Technology, Internet Appliance, Smart Homes .*

## I. INTRODUCTION

One of the most recurrent visions for humanity has been the possibility of having smart houses managed by computers or robots that undertake all kind of activities in order to support an easy and pleasant life. Nowadays, due to the advances in computer science and technology, it seems that we are close to turning the intelligent environments' vision into reality.

This raises the question of what an intelligent environment is. An Intelligent space (or iSpace) is "the world where information and communication technology (ICT) disappears as it becomes embedded into physical objects and the spaces in which we live and work" [1] [2].

An intelligent environment might be perceived as a number of electronic devices connected by a network, exchanging information and working together to orchestrate the operation of an environment in a holistic manner. Examples of this concept are the digital homes, smart homes and the intelligent buildings, in which most of the devices interact in the physical

world with the inhabitants, to coordinate and automate actions for specific situations [3].

According to Steventon, an intelligent space is formed mainly by a) the user's physical world, b) the interface between the digital world and the physical world (which includes hardware, software or both) and finally c) the digital world which supports actions in the physical world. In his view the purpose of an intelligent environment is to provide comfort to the inhabitants by allowing them to automate behaviours or complete routine tasks more easily and even to anticipate any of the inhabitant's requirements [1].

To encourage the inhabitants to interact with this environment without any kind of specialised knowledge, the interaction between the users and the environment should be as transparent as possible, through intuitive and user-friendly interfaces. Diverse approaches to these issues have motivated vigorous research, resulting in different kinds of user-interfaces each seeking to allow a more intuitive relationship between the intelligent environment and the user. Some examples are the voice-command, gesture recognition and web interfaces or even more experimental approaches such as touch [4].

## II. PERVASIVE INTERACTIVE PROGRAMMING

A new approach to addressing this problem is the Pervasive Interactive Programming (PiP), which is a computational framework introduced by the authors, and it can be viewed as a variant of the Programming-by-Example paradigm [5].

The main objective of the PiP model is to allow non-technical people to construct novel composite collections of coordinating embedded-computer based network services, by demonstrating the required system behaviours, using real home based appliances (see figure 1). These collections are called virtual appliances. [6]

### A. Meta-appliances

The data object representation of a virtual appliance is called a Meta-appliance (MAp). MAps define the membership of 0-n services, provided by the physical devices, and the behaviour, related to 0-n rules established for the virtual appliance which takes into consideration the user who creates it. The rules contain two parts, the conditions or "antecedents" and the actions or "consequents". A rule in PiP is a combination of one or more antecedent and one or more

consequents. The actions of one rule are only executed if the conditions of the rule are satisfied.



Figure 1. Creating MAs via physical interaction in the Essex iSpace

### B. Architectural Model

Figure 1 shows conceptual architectural framework of PiP. It consists of the modules described below [6].

1) *Graphical interface*: The graphical interface refers to the module that allows the user to interact with the environment.

2) *MAp Management module*: This module is responsible for managing the MAs previously created or owned by the user and keeps track of which MAs are currently being executed.

3) *Login module*: The Login module identifies the current user. If a user is not registered PiP only allows access to certain functions.

4) *Rules Management module*: This module manages the rules of the current MAp and the formation of new rules during a “programming-by-example” session. This is based on event notifications originated by the user interactions on either the GUI or the physical environment.

5) *MAp Execution module*: This module executes the MAs according to rules previously established.

6) *Knowledge Bank*: The Knowledge Engine module acts as a repository for MAs and other PiP system information. MAs are based on the Decomposed Community Programming (dComp) ontology whose most important feature is the concept of community. Each entity (e.g. device, services, community etc) that exists in the dComp environment is referred to as document which describes a complete list of classes, properties and relationships of network entities in a machine and human understandable format. [7]

7) *Eventing module*: The Eventing module manages the network notifications and relates them to the objects (eg switching on a light) allowing communication between parties.

8) *Interaction Execution Engine*: This module acts as a network control point, executing device discovery and sending networked service requests via UPnP on top of TCP/IP protocols.

9) *UPnP network*: UPnP is a set of event-based protocols for networking, used in this model for service discovery and communication within the home.

### C. Web GUI Motivation

Although the PiP model encourages personalization of the environment by means of the user physically demonstrating their desired behaviour, it also established the necessity for having a secondary GUI in order to programme actions with variables that can not be used as physical interactions e.g. the temperature, time etc.

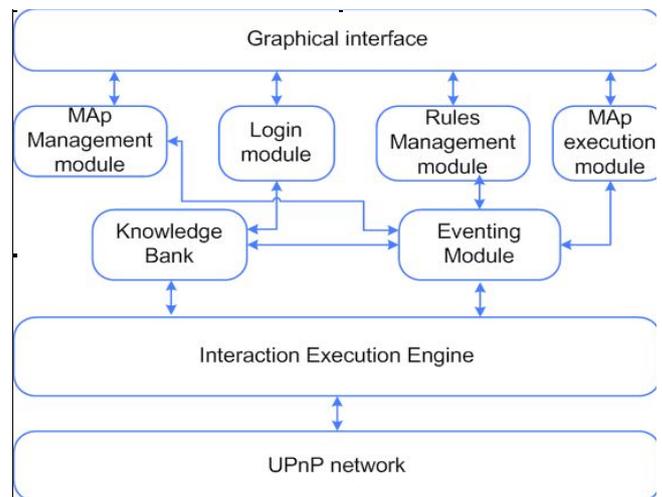


Figure 2. PiP Original Architecture.

### III. ENHANCING PiP WITH A WEB BASED GUI

In addition to the need to be able to provide such non-physical parameters, there is also the need to consider that real environments are not structured only for the sake of a single person but sometimes for a hierarchy of people that could be very complex with multiple levels of authority and participants (e.g. large companies) or simple with just one level (e.g. nuclear families). In order to solve this problem it is crucial to establish a structure that supports several users with different kinds of privileges. At the heart of this is the notion of secure logins which identify and associate users to their own environments (and bar associations to unauthorised people). This model of user-profiles and user-privileges is also useful for making ‘virtual appliances’ portable, such as when sharing a ‘virtual appliance’ between a home and office. Finally, this model would also facilitate sharing or trading virtual appliances between different users by providing a concept of ownership, or authorship to MAs. Finally, to simplify the process of programming, intuitive icons

representing entities (sensors, actuators, appliances etc) and processes (programming constructs) would be especially useful.

An effective solution these challenges would be to provide PiP with a web-based Graphical User Interface (GUI) to interact with an intelligent environment in order to create their own virtual appliances. In doing this, it is necessary to keep in mind that this proposed application must be completely focused on the end-users, providing a user-friendly way to integrate and share virtual appliances. By making this interface web based, the proposed GUI may be published on the Internet and be accessed in this way by any computer or person, which has several advantages. One is to allow different non-technical developers to use the model. Another is that the application can be accessed remotely. For example the user will be able to change the virtual appliances rules when they are away from home, or to be informed of any change in the current situation of the environment. Besides, the interface could be used to expand the possibilities for experimental research by, for example, facilitating the acquisition of data regarding the habits of different kinds of users (socio-economical levels, cultural levels, etc.) thereby widening the field of research.

#### A. A Web Argumented PiP Architecture

In addition to the many virtues, there are negative aspects to consider such as that the application, and thus the user-data, is exposed on the internet, and it is therefore important to implement security measures to reduce the possibility of compromising privacy or security, e.g. criminal activity or industrial espionage.

Figure 2 shows how we have augmented PiP with a distributed web service scheme and utilised a GUI based on a Rich Internet Application (RIA). RIAs are web applications that use rich media resources (audio, video and graphics) to improve the user-experience. These applications offer several advantages ranging from providing a more user-friendly environment to improving the performance, by eliminating some data transfers between the server and the clients with the help of a plug-in or a virtual machine [8].

W3C defines a "web service" as "a software system designed to support interoperable machine-to-machine interaction over a network" [9]. In accordance with this concept, we have modified the PiP Architecture to include a new layer of web services that can connect any PiP based environment to any web browser including regular computers and mobile phones. The Web augmented section of the PiP Architecture is described below.

1) *Security module*: The original Login module was modified to include a user privileges schema which manages a users ability to perform actions based on permissions pre-assigned to the user. In addition it must authenticate and authorize the users to connect to his/her PiP environments only.

2) *Web Services*: The web services layer includes all the processes needed to verify the identity the user and to create and manage MAs which interact with the UPnP network to create virtual appliances. The implementation is based on SOAP Web Services rather than REST based due to the large amount of information that is sent. Security and integrity of the data is important so, with the addition of this new layer, we have integrated a security mechanism to authenticate users

3) *RIA Web, Mobile Telephone & Other GUIs*: This layer includes all RIA applications that can be connected to one or more environments by means of web services. The interfaces included in this module manage the users' profiles, providing a customisable application capable that emulate's the user's real digital home environment, allowing a more intuitive user interaction.

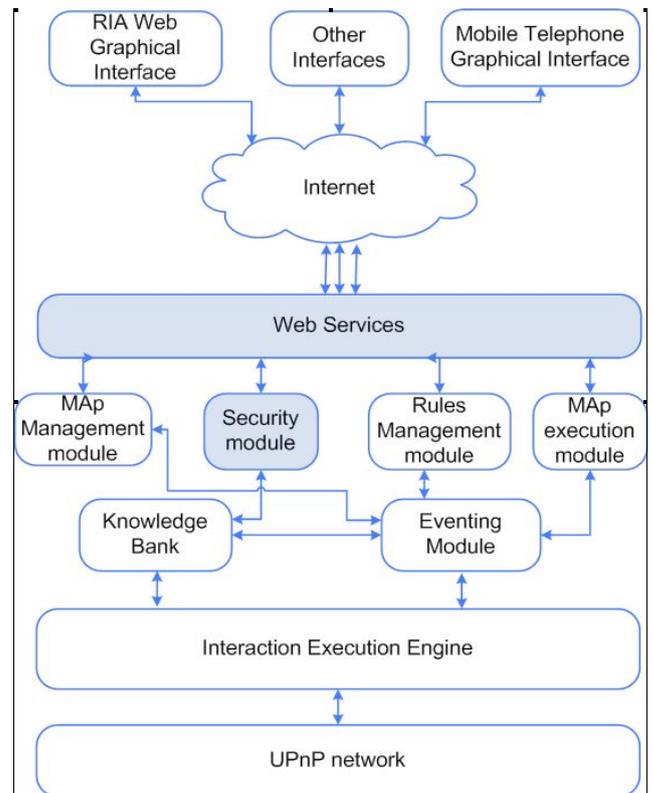


Figure 3. Web Augmented PiP Architecture

The framework we are using to create the RIA web-based application is Java FX, which is an open source java-based framework for cross-screen RIAs. It provides rich media and device specific components for desktop, mobile, and media devices such as televisions.

Java FX offers many advantages such as portability by running a Java virtual machine together with providing an efficient software life-cycle paradigm based on object oriented programming. Another advantage is better security over other RIA frameworks (e.g. Ajax). Finally Java FX allows the possibility of creating complex and user-friendly RIAs.

Thus Java FX offers the means to develop an intuitive, flexible and secure GUI for PiP that allows users to manage and share virtual appliances between them and their environments.

The following fictional scenario illustrates how the above model supports PiP programming and management.

*“It is a new day at the elementary school. Just a few minutes before the classes start the children arrive and walk from the halls to their classrooms. The sound system plays some lively music. Sarah is a teacher in this school. She always arrives before her students and the first thing she usually does is to login to the school IT environment via a web application. Once she is authenticated by the system, she organise her files for the today’s class. At 9:00 o’clock the bell rings indicating that the class must start. At that moment the lively music in the sound system decrease until it is turned off and the computer system starts the lesson selecting the first file which was selected by the teacher minutes previously. Earlier that morning, before coming to the class, Sarah prepared herself a light meal at home and now, in the middle of the lesson she had a worrying thought that she may not have turned off the hob! While the students were watching a video for today’s history class she takes a moment to connect her to home via the web application which, after validating her credentials, shows a virtual representation of her kitchen in the screen of her laptop. She asks for the appliances’ status in the kitchen. Everything is ok; the hob was turn off. Maybe her husband noticed and turned off the hob. She remembers that one of her colleagues told her that she programmed her house to turn off all the kitchen appliances when there is no adult in the house. Later she will ask her colleague for this MAP, so she can load it into her home and modify the MAP so that it can be valid only for herself, but now she must focus on the class again”.*

As it was described in the fictional scenario, this augmented architectural model facilitates one or more users connecting to the same intelligent environment (in the case of the fictional scenario every teacher in the school must log into the application), supports different user environments (e.g. the teacher’s school and home environment) where each may be personalised by individual users in different ways (for example the modified rule for the kitchen appliances which will be valid for the teacher but not for her husband) and provide an easy way to share virtual appliances (e.g. the colleague’s shared MAP). The architecture also includes a security module that manages logins, including hierarchy control, such as allowing users to deny or accept privileges (for example the colleague’s kitchen appliances rule is only valid when there is no adult in the house, ignoring children).

#### IV. SUMMARY AND FUTURE DIRECTIONS

In order to make the use of intelligent environments easier and more attractive to end-users, it is essential to develop more user-friendly modes of interaction. In this paper we have described how this can be achieved by augmenting a well-proven end-user programming paradigm, the Pervasive interactive Programming (PiP) architectural model, with a

distributed web service model that utilises a GUI based Rich Internet Application (RIA) framework. Towards these ends, this work-in-progress paper describes a marriage of a well established and proven architecture, PiP, with a more conceptual distributed web service layer, Java FX RIA, that increases the flexibility of PiP by giving access to multiple and hierarchical web based users. Whilst this approach will bring many advantages we acknowledge that there are also difficult challenges, such as ensuring the security and privacy of users on the Internet. Clearly, there will be many lessons to be learnt from this work as it progresses, which we hope to report in subsequent conferences.

#### ACKNOWLEDGMENT

We are pleased to acknowledge Anasol C Peña Ríos sponsors, the Human Resources Development Fund (FIDERH) of Banco de México and the Santander Universities Network for UK, Latin America and Portugal.

#### REFERENCES

- [1] A. Steventon and S. Wright (Eds.), “Intelligent Spaces: The Application of Pervasive ICT”. Springer-Verlag compilation. London, pp. xv-xvii, 1-17, January 2006.
- [2] M. Weiser “The Computer for the 21st Century”, Scientific American, Vol. 265, No. 3, pp.94-104, September 1991.
- [3] B.D. Johnson, V. Callaghan, M. Gardner; , “Bespoke appliances for the digital home”, IET 4th International Conference on Intelligent Environments, Seattle, USA, pp. 1-7, 21-22 July 2008.
- [4] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, J. S. Y. Chin, “An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning”, Eight International Conference on Ubiquitous Computing (UBICOMP 2006), Irvine, California, USA, Sept 2006.
- [5] J.S.Y. Chin, V. Callaghan, G. Clarke, “Soft-appliances: A vision for user created networked appliances in digital homes”, Journal of Ambient Intelligence and Smart Environments, pp. 69–75, 2009.
- [6] J.S.Y. Chin, “Pervasive Interactive Programming: Empowering End Users to Customise Digital Homes”, Ph.D. Thesis, School of Computer Science and Electronic Engineering, University of Essex, pp. 144, June 2009.
- [7] J.S.Y. Chin, V. Callaghan, M. Colley, H. Hagrais, G. Clarke, “Virtual appliances for pervasive computing: A deconstructionist, ontology based, programming-by-example approach” in IEEE International Workshop, Intelligent Environments 2005 (IE05), Colchester, UK, 28-29 June 2005.
- [8] J. C. Preciado, M. Linaje, F. Sanchez, S. Comai, “Necessity of Methodologies to Model Rich Internet Applications”, in ‘WSE ’05: Proceedings of the Seventh IEEE International Symposium on Web Site Evolution’, IEEE Computer Society, Washington, DC, USA, pp. 7–13, 2005.
- [9] World Wide Web Consortium (W3C), “Web Services Architecture”, February 2004. Online at: <http://www.w3.org/TR/ws-arch/#whatis> Accessed February 2010.
- [10] J. M. Silveira. Neto, Sun Microsystems, “JavaFX Overview”, CEJUG Event, Brazil, 2008. Online at: <http://www.slideshare.net/silveiraneto/javafx-overview-presentation> Accessed April 2010.