

End User Tool for Deploying Smart Object Systems

Fahim Kawsar and Tatsuo Nakajima
Department of Computer Science, Waseda University
Tokyo, Japan
{fahim,tatsuo}@dcl.info.waseda.ac.jp

ABSTRACT

We present a deployment tool atop a document centric infrastructure to support end users in constructing and extending smart object systems. This tool allows an ordinary individuals to deploy, extend and control smart object systems in a Do-It-Yourself (DIY) fashion. We have implemented two versions of this tool using different user interfaces. The first version provides a graphical user interface for the deployment tasks whereas the second version provides a tangible user interface. The tangible tool allows end users to deploy and control artefacts and applications only by manipulating RFID cards that comes with the artefacts and the applications. An underlying infrastructure provides the foundation for the end user deployment utilizing a profile based artefact framework for smart objects and a task oriented application framework for pervasive applications. A runtime component, FedNet associates the application to corresponding smart objects. In this demo, we will showcase this tool atop our infrastructure to construct two everyday smart object systems with a broad goal of identifying the applicability and usability of such tool in our everyday life.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Interaction Styles

Keywords

Smart Object, Deployment, Graphical and Tangible User Interfaces

1. INTRODUCTION AND APPROACH

We envision that the convergence of ubicomp technologies will eventually bring smart object systems into our home. A smart object system usually involves sensors, augmented artefacts, mobile devices, displays, etc. Ideally, these systems should be similar to the home appliances, i.e., easy to setup, adaptive to users' needs, and interchangeable with new models. A user should buy one or multiple physical artefacts and applications for them and be able to install them just like other home appliances. In addition, one can incrementally enhance the application's functionalities by upgrading the artefacts' features. Involving end users in the deployment

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiQuitous 2008 July 21 - 25, 2008, Dublin, Ireland.
Copyright © 2008 ICST ISBN 978-963-9799-27-1.

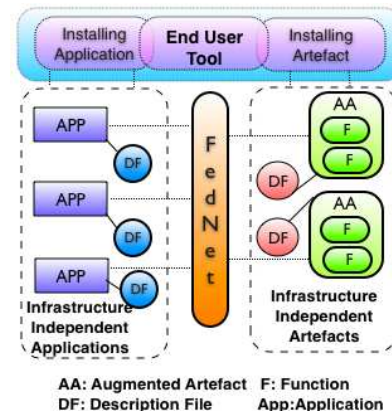


Figure 1: Basic workflow of our approach

process requires the development of plug and play artefacts that are deployable in a *Do-It-Yourself (DIY)* fashion. In addition, the deployment process of these systems has to be seamless without complex configuration and administration.

To address these concerns, we have developed a document based artefact integration infrastructure that provides the foundation for the end user deployment utilizing a profile based artefact framework and a functional task oriented application framework [2]. The artefact framework represents an instrumented artefact by encapsulating its augmented functionalities in one or multiple service profiles atop a core (generic binary), expresses these services in generic documents and allows additional profiles to be plugged into the core incrementally. Such generality makes an artefact plug and play and allows end users to deploy it easily in a DIY fashion. Applications in our approach are represented as a collection of functional tasks independent of their implementation in a corresponding document. These tasks are atomic actions that require artefacts' services, e.g., "sense current humidity", "turn on the lamp", etc. A infrastructure component FedNet, manages these applications and artefacts and maps the task specifications of the applications to the underlying artefacts' services by matching respective documents (that express the applications and the artefacts) to create a spontaneous federation. Figure 1 shows the basic functional workflow of our approach.

The DIY deployment process of a smart object system requires two tasks: i) Installing a ready-to-run instrumented artefact or attaching sensors or actuators to an existing artefact, and ii) Installing applications and associating the application with artefacts when necessary. The attachment of sensors or actuators requires active



Figure 2: Snapshots of GUI Tool (a) Main Panel, (b) Installing Application (c) Adding Profile

guidelines so that the installation is successful. Each of the artefacts in our approach has a proxy component that represents the artefact digitally. This artefact proxy is disseminated as a generic binary. The artefact profiles implementing the augmented features are self-contained plug-ins that can be installed atop the generic binary. Applications are also released in executable binaries in our approach. These binaries can be installed by the end users just like any other executable softwares (i.e. double clicking the executable, and following the on-screen instructions). However, in our approach these installation mechanisms must reside within the FedNet environment, so that FedNet can provide the spontaneous federation among the applications and the artefacts. Since, in a smart object system, one application can use multiple smart objects, one important deployment task is to associate an application with one or multiple smart objects. The final end user task is controlling (running and stopping) a smart object system. To support these tasks we have developed a deployment tools for end users. The basic functionalities of the tool are as follows:

1. *Install/Uninstall an Artefact/Application*
2. *Adding/Removing a Profile to an Artefact*
3. *Associating an Application with Artefacts*
4. *Runing/Closing an Artefact/Application*

We have developed two versions of this tool. The first version is a web based Graphical User Interface (GUI). However our pilot user study revealed several usability problem of GUI based approach towards deployment activities. Hence, we developed the second version of our tool using a tangible user interface. In the following we provide a brief summary of these interfaces.

1.1 GUI Tool for Deployment Tasks

Figure 2 shows some snapshots of the web based GUI tool that supports the deployment tasks. End users can perform all the four tasks mentioned above. Typical activities include adding artefacts or applications in the FedNet environment by uploading the executable binaries, adding profile to artefacts by uploading the binaries and attaching the hardware to the artefact following the on-screen instructions and associating applications to artefacts by creating virtual links between them (drag-drop metaphor). End users can also start and stop a system using the GUI.

1.2 Tangible Tool for Deployment Tasks

Our pilot user study of GUI tool revealed several usability issues. In our findings, primarily the problem was with the end users' conceptual model of installing home appliances. Although, the deployments tasks were identical to regular desktop computing, e.g., software installation, the end users found it difficult to comprehend

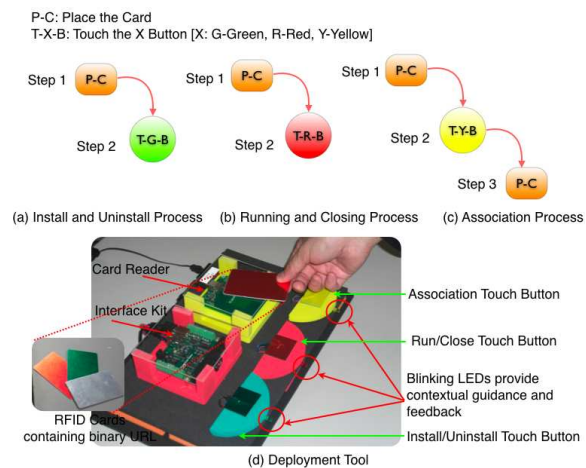


Figure 3: (a-c) Deployment Steps (d) End User Deployment Tool

because of the presence of the physical artefacts and suggested that the process has to be more mechanical and tangible. Following their suggestions we have adopted a tangible user interface for the deployment purpose. RFID (representing artefacts, application and profiles) and RFID Reader with touch buttons are provided for end users' interaction. Figure 3 shows the tangible tool for the deployment tasks and its interaction mechanism. Each of the artefacts, profiles and application is disseminated in a generic binary in a corresponding RFID embedded card. The end users can install/uninstall, run/stop artefacts and applications by using these cards and the deployment tool. For example, installation of an artefact or application can be performed by placing the card of the artefact/application onto the tool and then pressing the green button (Figure 3(a)). Furthermore, they can associate a profile or an application to a specific artefact or a group of artefacts. The tool provides audio and visual (LEDs) guidelines to assist these tasks.

2. DEMONSTRATION

In this demonstration, we would like to showcase the deployment tool to construct and extend functionalities of two *proof-of-concept* smart object systems. The first system uses an augmented mirror to show some personalized information proactively[1]. The second system has the objective of improving users' dental hygiene by promoting correct toothbrushing practices [3]. The application turns a mirror into a simulated aquarium. Fish living in the aquarium are affected by the users' toothbrushing activity. In the demonstration session we would invite the participants to deploy these systems and to extend their functionalities.

3. REFERENCES

- [1] K. Fujinami, F. Kawsar, and T. Nakajima. Awaremirror: A personalized display using a mirror. In *Third International Conference on Pervasive Computing*, 2005.
- [2] F. Kawsar, K. Fujinami, and T. Nakajima. A document centric approach for supporting incremental deployment of pervasive applications. In *The 5th Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2008)*, 2008.
- [3] T. Nakajima, V. Lehdonvirta, E. Tokunaga, and H. Kimura. Reflecting human behavior to motivate desirable lifestyle. In *The Conference on Designing Interactive Systems (DIS 2008)*, 2008.