EXPERIENCING EXTROVERT GADGETS

Irene Mavrommati† Panos Markopoulos‡, John Calemis†, Achilles Kameas†

†Research Academic Computer Technology Institute
R. Feraiou 61, Patras, Greece
{Irene.Mavrommati, Calemis, Achilles.Kameas}@cti.gr

‡Technische Universiteit Eindhoven
Den Dolech 2, 5600MB Eindhoven
The Netherlands
P.Markopoulos@tue.nl

ABSTRACT

This paper discusses the concepts and the supporting infrastructure developed within the extrovert-Gadgets research project. This project explores concepts and technologies that aim to enable end-users to realize, modify and personalise Ubiquitous Computing applications. The eGadgets technology is at a proof of concept stage and the demonstration presented shows the feasibility of augmenting physical objects with a digital 'self' and for defining the combined behaviour of a few such augmented artefacts through a purpose made editor.

Keywords

Ubiquitous computing, end-user programming, component architecture, GAS architectural style.

1. INTRODUCTION

Our future environments will be furnished with an increasing number of computationally augmented artefacts. These may be not only information appliances, but also ordinary objects enhanced with computing and communication capabilities. In such a technological landscape people may find themselves surrounded by artefacts, which are distinct yet interconnected via a web of network services. The approach presented in this paper is to create an appropriate infrastructure and tools, to enable people to make their own applications with 'augmented' artefacts.

Extrovert-Gadgets (eGadgets) is a research project that is part of the EU-funded Disappearing Computer initiative (http://www.disappearing-computer.net). The project extends the notion of component-based software architectures to the world of tangible objects, thereby transforming objects in peoples' everyday environment into autonomous artefacts (the eGadgets), which can be used as building blocks of larger systems. The ubiquitous computing environments formed by such artefacts are intended to be accessed directly and to be manipulated by untrained end-users.

The paper presents the concepts, infrastructure and tools that have been developed within the eGadgets research project, describes the demonstration scenario and summarises a preliminary assessment of the user considerations relating to the proposed concepts.

2. CONCEPTS AND TERMINOLOGY

A vocabulary of basic terms acts as a common referent between people, objects and their collections. These are the following:

- An eGadget is defined as an everyday physical object enhanced with sensing, acting, processing and communication abilities. Moreover, processing may lead to "intelligent" behaviour, which can be manifested at various levels.
- Plugs are software classes that make the eGadget's capabilities visible to people and to other eGadgets.
- Synapses are associations between two compatible Plugs (figure 1). People can utilize Plugs in order to form Synapses (and thus create GadgetWorlds). Synapses are effected through middleware operating over a wireless network and are only conceptualised by users as invisible links.
- A Gadgetworld is a functional configuration of associated eGadgets, which collaborate in order to realize a collective function.
- The Gadgetware Architectural Style (GAS) constitutes a generic framework, shared by users and designers, for consistently describing, using, reasoning about GadgetWorlds. GAS defines the concepts and mechanisms for people to define GadgetWorlds out of eGadgets.
- A Gadgetworld Editor (GE) is a facilitator device that can be used for Gadgetworld creation and editing. The purpose of the Editor is threefold: a) to indicate/make visible the available eGadgets and Gadgetworlds b) to form new Gadgetworlds c) to assist with debugging, editing, servicing, etc.

GAS is implemented in GAS-OS, which is the middleware that enables the composition of GadgetWorlds as distributed systems [3]. GAS-OS is a component framework [4] that manages resources shared by eGadgets, it determines their software interfaces and it provides the underlying mechanisms that enable communication (interaction) among eGadgets. The proposed concept

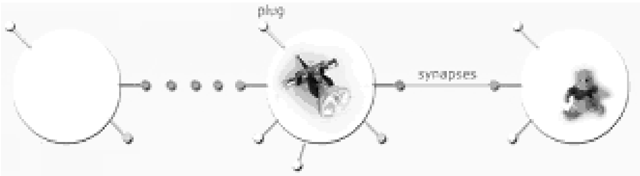

Figure 1: A visualisation of the basic vocabulary terms: eGadget, Plug, Synapse, GadgetWorld

supports the encapsulation of the internal structure of an eGadget (eGadgets are treated as “black boxes”, based on their public interface as manifested by the Plugs), and provides the means for composition of an application, without having to access any code that implements the interface [2]. Thus, this approach provides a clear separation between computational and compositional aspects of an application [4], leaving the second task to end-users. The benefit of this approach is that, to a large extent, system design is provided ready to the end user, because the domain and system concepts are specified in the generic architecture.

Composition is effected through the definition of synapses between pairs of plugs. Each end of a synapse is managed by the GAS-OS running on each eGadget. So, in order to have, e.g., a coffee maker prepare coffee when the alarm sounds, one has to connect the Alarm Output Plug of a clock eGadget to the Switch on/off Input Plug of the coffee maker. A synapse then serves as the abstraction of a communication channel between peers. However, this occurs only when they have ‘discovered’ each other. Discovery may be forced upon the eGadget by a user creating the synapse with the editor, or proactively carried out by an eGadget (for example, when the coffee maker in the kitchen breaks down, the clock may look for another coffee maker in the house). Nevertheless, a synapse is still a logical abstraction: physically, the Plug/Synapse model is implemented with a peer to peer architecture, where the GAS-OS of the two eGadgets participating in a synapse will exchange events and data in a way specified by the adopted communication protocol.

In order for two plugs to form a synapse, they have to be compatible. Compatibility is a very broad issue not completely resolved within the project - for the moment, we implement data type compatibility and value range compatibility.

There exist two types of Plugs in every eGadget. One Tplug, which is the commonly available interface, and several Splugs, which represent the services offered by the eGadget. The Tplug contains the "public" eGadget information, that is ID, brand, location, physical properties (i.e. shape, colour etc), available services (i.e. list of Splugs), synapses and their state, etc. It can be queried at a commonly known address for all eGadgets.

To the point of writing the paper, 10 sample eGadgets have been created, as a test implementation of embedding the proposed platform into everyday objects. A software tool, the GadgetWorld Editor, was created to facilitate the composition of GadgetWorlds. Two versions of the editor have been created. One with richer functionality runs on a laptop personal computer and is intended for the eGadget ‘professional’ designer. The second and simpler one runs on an iPAQ handheld computer and is intended for the non trained end-user.

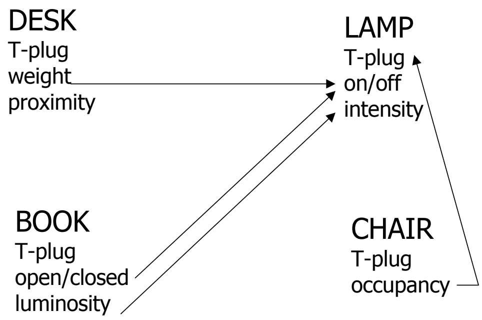

Figure 2. Graphic visualisation of a mini-gadgetworld aiming to switch on the light when a person sits in front of a desk and opens a book. The synapses are illustrated as directed links, showing the dependencies between eGadget behaviours.

Using the handheld editor users can purposefully associate the Plugs of different eGadgets and synthesize GadgetWorlds as ordered sets of Synapses. As eGadgets can communicate and interact, peoples’ environments exhibit a highly dynamic behaviour. Currently the project is experimenting with intelligent mechanisms that aim to learn from people’s use of a Gadgetworld and (transparently) optimise it and ease the formation of GadgetWorlds.

2.1 Example of GadgetWorld

Lets assume a story involving the use of everyday artefacts: John 21 is familiar with PC use, (text editing, web-searching). John has recently created his Study Application (a simple ubiquitous computing environment), with a new “extrovert-Gadgets” system he recently bought. He has set up his Study Environment to turn on the light automatically when he is studying on his desk, since the desk light switch is at the back of the shelves and it is hard to reach it. This functionality can be supported by a GadgetWorld consisting of four eGadgets: a desk, a desk-lamp, a book and a chair (Figure 2).

The eGadgets in this example have plugs through which they offer access to their properties: for example, the book is equipped with a light sensor whose reading is made available through the plug ‘luminosity’. When the lighting is lower than a certain threshold, someone sits on the chair and the book is open and on the desk, the synapse with the on/off switch of the lamp will make the lamp to switch on.

3. Demonstration Scenario

A proof of concept demonstrator has been created and will be shown at the conference. The demonstration will illustrate the steps for setting up a GadgetWorld.

First, the GadgetWorld editor, which is also an eGadget, is started up. This editor runs on an iPAQ computer (Figure 3). Through wireless communication and by use of the GAS-OS middleware, the editor finds other eGadgets in the network vicinity and displays their names on the screen (see [2] and [3] for a more detailed technical description).

In this demonstration the Gadgetworld editor will find::

- A MATHMOS lamp converted to an eGadget (MATHMOS lamps resemble luminous bricks that illuminate in different colours depending on which side they are standing).
- An MP3 player, which for the purposes of the demonstration will be the WinAmp running on a laptop.
- An eCarpet, which is a small mat equipped with a pressure sensitive grid.

All the above objects are GAS compatible. Their digital self offers their capabilities through their plugs, that are listed on the editor display as a drop down list associated with each eGadget. In the demonstration these plugs are associated to each other and synapses are created so that several functions can be demonstrated step by step:

The demonstration will show the following:

- On the GadgetWorld editor we inspect the plugs offered by the eGadgets. The MATHMOS eGadgett has only one plug, the "side" plug, which knows on which side it has been placed. The mp3 player has several plugs associated with it, i.e.: a pause, play, track number and music gender plugs.
- Through the GadgetWorld editor we create a "synapse" between the carpet and the MP3 player and between the MATHMOS and the MP3 player.
- We turn the MATHMOS on its side and a different type of music will play.
- Stepping at different areas of the mat causes the MP3 player to play, pause and the volume to change.

4. Evaluation

A central research question for the eGadgets project is whether the end-user will be inclined and able to use GAS

Figure 3. The GadgetWorld Editor running on an iPaQ.

to serve his/her needs. To address this issue, a formative evaluation of the concepts has been carried out. The evaluation was conducted in two parts. First an expert review was conducted in the form of a workshop. Subsequently, the cognitive dimensions framework [1] was applied to assess how well the eGadgets concepts support end-users to compose and personalise their own ubiquitous computing environments.

The evaluation workshop involved three experts in user system interaction. The end-user of eGadgets is considered to be a 'technophile' but not a programmer; the three experts matched this profile. During this session a collection of four scenarios was discussed that highlighted different usage/interaction design issues. Experts were given a problem solving exercise, for designing their own GadgetWorld on paper and during which they could comment on anticipated usage and acceptance problems.

The Cognitive Dimensions framework [1] is a broad-brush technique for the evaluation of information artefacts, e.g., notations and interactive systems. It helps expose trade-offs that are made in the design of such information artefacts with respect to the ability of humans using them to capture their concepts and intentions and to manage and comprehend the artefacts they create. In broad terms, GadgetWorlds can be perceived as computer programs which are composed from pre-defined modules in non-textual manner. The Cognitive Dimensions framework was used to assess alternative choices with respect to the primitives offered by eGadgets, the abstraction level targeted and the functionality of the tools for GadgetWorld construction.

The feedback to the project, is captured as four tentative design principles [5]:

- The Gadgetworld behaviour should not surprise the user, i.e. automation or adaptation actions should be visible and predictable (or at least justifiable).
- Simple tasks should remain simple even in an intelligent Gadgetworld. Intelligence should be applied to simplify complex tasks.
- End-users acting as Gadgetworld developers should be supported with at least as good tools as programmers have at their disposal, providing functionality that corresponds to debuggers, object browsers, help, etc.
- Multiple means to define user intentions should be supported by the graphical editor, as the users tasks tend to be comprehended and expressed in a variety of ways, e.g., in production rules, patterns, procedures, etc.

5. FUTURE WORK

With eGadgets we have set out to enable people to actively shape ubiquitous computing environments. Concepts have been borrowed from component based software development and applied to convert physical objects into components of Ubiquitous Computing environments.

The eGadgets project aims to address the needs of two broad classes of users. The professional designer creator of Gadgetworlds that may be sold to the public and the end-user/consumer of such products. So far, we have considered most the needs of the end-user, which is the most crucial for the success of our concepts. In the remainder of the project, two evaluation activities are planned. Each addresses one of the two user categories defined above:

To assess the longer term use of eGadgets in an ecologically valid manner, a field experiment shall be conducted at the i-Dorm of the University of Essex (iieg.essex.ac.uk/idorm.htm). A very small number of test-users will have to create their GadgetWorld in the room they will be staying for at least one overnight stay. The intention is that they shall get over the initial impression (whether positive or negative of the technologies proposed) and experience the advantages or disadvantages of inserting eGadgets in their daily lives.

To assess how designers outside the research consortium can use the infrastructure provided, trainee designers will have to create their own eGadgets to create a smart carpet, within a period of 13 weeks. We intend to keep journals of this activity and report on the practical and conceptual problems they encounter.

6. REFERENCES

[1] Green,T.R.G., Petre, M. (1996). Usability analysis of visual programming environments: a cognitive dimensions framework. Journal of Visual Languages and Computing. *J. Visual Languages and Computing*, 7, 131-174.

[2] Kameas, A., Ringas, D., Mavrommati, I., and Wason, P., (2002) an Architecture that Supports P2P Networking Among Ubiquitous Computing Devices, IEEE P2P 2002 Conference, Linkoping, Sweden, Sept. 2002.

[3] Kameas A., Bellis, S., Mavrommati I., Delaney K., Colley M., Pounds-Cornish A. (2003) An Architecture that Treats Everyday Objects as Communicating Tangible Components, in Proc. of the 1st IEEE International conference on Pervasive Computing and Communications (PerCom03), Forth Worth, USA, March 2003

[4] Schneider. J.G., Nierstrasz O. (1999), Components, scripts and glue. In Software architectures - advances and applications (J. Hall and P. Hall - eds), Springer-Verlag, pp 13-25.

[5] Mavrommati I., Kameas, A., Markopoulos P., (in press) Visibility and accessibility of a component-based approach for Ubiquitous Computing applications: the eGadgets case. To be presented at HCI International, June 2003, Creta, Greece.

[6] Kameas A, Mavrommati I. (in press) End-user programming tools in ubiquitous computing applications, To be presented at HCI International, June 2003, Creta, Greece.