Visibility and accessibility of a component-based approach for Ubiquitous Computing applications: the e-Gadgets case.

Irene Mavrommati

Research Academic Computer Technology Institute
R. Feraiou 61,
Patras, Greece
Irene.Mavrommati@cti.gr

Achilles Kameas

Research Academic Computer Technology Institute
R. Feraiou 61,
Patras, Greece
Achilles.Kameas@cti

Panos Markopoulos

Technische Universiteit Eindhoven
Den Dolech 2
5600 MB Eindhoven
the Netherlands
P.Markopoulos@tue.nl

Abstract

The paper firstly presents the concepts and infrastructure developed within the extrovert-Gadgets research project, which enable end-users to realize Ubiquitous Computing applications. Then, it discusses user-acceptance considerations of the proposed concepts based on the outcome of an early evaluation.

1 Introduction

Extrovert-Gadgets (e-Gadgets) 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 development 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.

eGadgets (www.extrovert-gadgets.net) have a tangible self and a software self. They range from simple objects (e.g. lights, switches, cups) to complex ones (e.g., PDAs, stereos) and from small ones (e.g., sensors, pens, books) to large ones (e.g., desks, rooms, buildings). It is intended that a lay person (i.e., who does not have software development skills), can actively shape his/her environment by associating eGadgets into collaborating functional collections (the GadgetWorlds). Such a system can be perceived and analyzed from the standpoints of interaction, which happens at different levels: between eGadgets, between people and eGadgets, or between people and GadgetWorlds. The paper presents the concepts and the infrastructure that has been developed within the e-Gadgets research project, and further it discusses a preliminary assessment of the user acceptance considerations relating to the proposed concepts.

2 Concepts and infrastructure

In the project's approach a vocabulary of basic terms acts as a common referent between people, objects and their collections (Figure 1). These are the following:

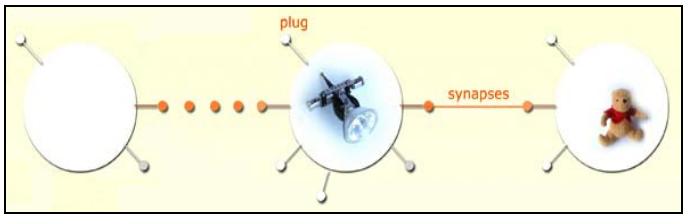

Figure 1: A depiction of the basic vocabulary terms: eGadget, Plug, Synapse, GadgetWorld

- An *eGadget* is defined as an everyday physical object enhanced with sensing, acting, processing and communication abilities. Moreover, processing may lead to "intelligent" behavior, which can be manifested at various levels.
- *Plugs* are software classes that make visible the eGadget's capabilities to people and to other eGadgets.
- *Synapses* are associations between two compatible Plugs. People can utilize Plugs in order to form Synapses (and thus create GadgetWorlds). In this context Synapses can be described as the invisible links between the physical objects.
- 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.

The design of this novel hardware / software architecture supports computation with limited resources, ad-hoc networking and distributed resource sharing. The e-Gadgets project is developing a set of concepts defined within GAS and GAS-OS, which is the middleware that enables the composition of GadgetWorlds as distributed systems (cf. Kameas et al, 2002). A set of 10 sample eGadgets has been created, as a test implementation of embedding the proposed platform into everyday objects. Further, a software tool, the GadgetWorld Editor, was created to facilitate the composition of GadgetWorlds. People 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. Intelligent mechanisms are employed, aiming to learn from people's use of a eGadgetworld and (transparently) optimise it and ease the formation of GadgetWorlds. An eGadget is made of:

- A matrix of sensors and actuators and an FPGA-based board, which implements communication among the sensor board and the processor. It was decided to use the iPAQ as a platform to host the Java Gadget-OS and GAS-OS software because of the small size of these PDA modules. A separate FPGA based PCB was designed to complement the iPAQ and act as a programmable interface between sensors and the iPAQ. This PCB is in compact form 10cm x 10cm and allows up to 200 sensors actuators to be connected to the respective Gadgets. Communication between this board and the computational platform is through RS232. Two computational platforms are used, an IPAQ and a laptop, supporting WinCE. In some case, wireless transmission between the conditioning circuitry and the FPGA board is used in order to maintain user context.
- A processor module (including RAM and wireless module), which is currently served by an iPaq or a Laptop with the necessary wireless cards.
- The GAS-related middleware, which includes the following modules: Gadget-OS, which is specially implemented per every eGadget and used to control its resources; GAS-OS, which

manages the Plugs and Synapses; a communication module, which is responsible for implementing a discovery protocol and for establishing inter-eGadget communication; and eGadget GUI, which at the moment runs as a software simulation

The communication between the GAS-OS of two eGadgets is currently implemented using an XML-based messaging system. The complete software has been implemented in Personal Java.

3 Evaluation

A central research question for the e-Gadgets project is whether the end-user will be inclined and able to use GAS to serve his/her needs. An expert appraisal was carried out for evaluating the proposed interaction concepts and technology, with respect to the end user requirements. At the time of the evaluation, as most technology was still under development, a first version of the Editor was used, together with paper-mock ups of eGadgets. The nature of the evaluation was formative, i.e., it aimed to suggest directions for the next steps of the project, which would ensure that user needs are taken into account. Thus it did not focus on detailed interaction (look and feel) of the proposed GadgetWorld Editors, but on the concepts and on the role that e-Gadgets technology could play in fulfilling user needs. The evaluation evolved around two axes: Comprehensibility of the concepts and interaction and willingness to use such technology.

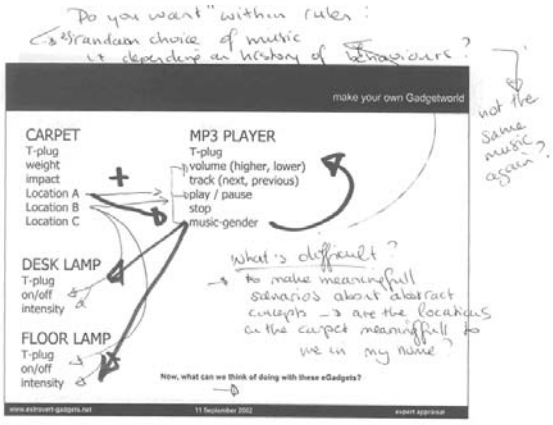

Figure 2. A GadgetWorld configuration made during the problem solving part of the evaluation.

The evaluation was conducted in two phases. First an expert review was conducted in the form of a workshop. Subsequently, the cognitive dimensions framework (Green and Petre, 1996) was applied to assess how well the e-Gadgets concepts support end-users to compose and personalise their own ubiquitous computing environments.

The evaluation workshop involved three experts in user system interaction and two of the authors. The end-user of eGadgets is considered to be a 'technophile' but not a programmer; the three experts matched this profile. The following activities were performed during the workshop.

- One of the authors introduced the basic concepts in a fashion similar to a seminar, in order to familiarize experts.
- A collection of four scenarios was discussed that highlighted different usage/interaction design issues. A small discussion in a focus group format followed each scenario.

- A problem solving exercise was set to gauge the extent to which these experts could build their GadgetWorld and further to reflect on what they consider as problems for the end-user.
- The GadgetWorld Editor interface and some video-prototypes for more advanced interfaces for constructing GadgetWorlds were demonstrated and expert opinions were solicited.
- An open-ended discussion elicited global level feedback for the project.

Some of the most recurring themes during the discussion are noted here:

- Ubiquitous computing technologies embedded in physical objects effectively add hidden behaviour and complexity to them. Problems may arise if this behaviour is not observable, inspectable and predictable for the user.
- Intelligence causes problems of observability and of unpredictability for users. It must be used with caution and this should be reflected in the demonstrations built.
- Constructing and modifying GadgetWorlds is a problem solving activity performed by end-users. As such, it has an algorithmic nature and thus good programming support should be offered[1].

3.1 Cognitive Dimensions

Following the last observation, one of the authors evaluated the GAS using the Cognitive Dimensions framework (Green and Petre, 1996), a broad-brush technique for the evaluation of visual notations or interactive devices. It helps expose trade-offs that are made in the design of such notations with respect to the ability of humans to translate their intentions to sequences of actions (usually implemented as programs) and to manage and comprehend the programs they compose. Broadly, GadgetWorlds can be perceived as applications (that is, complex programs), which are composed in non-textual manner. Thus, this theoretically sound technique can be used to provide insight into selecting between alternative choices with respect to providing tools for GadgetWorld construction. Below, we discuss some of the most interesting points resulting from this evaluation, along the dimensions defined in (Green and Petre 1996):

- Closeness of Mapping. eGadgets implement an architectural abstraction stemming from the nature of software composition, which is also close to the construction of real world objects, such as buildings and machines. However, due to the properties of the digital self of eGadgets, users might conceptualise their tasks in a variety of ways, such as stimulus-desired response, rules, sequences and constraints between entities, etc. In this way there will always be an initial gap between their intentions and the resulting functionality of a GadgetWorld, which people will have to bridge based on the experience they will develop after a trial-and-error process. A GW Editor can shorten this initial gap, by allowing several different ways of expressing the user's goals.
- Diffuseness/Terseness. At this stage e-Gadget appears to be diffuse, i.e., it has few conventions that need to be learnt.
- Hidden Dependencies. They are side effects, an issue well known to programmers: a state change in one component may have non-visible implications on the function of another. In the conceptual diagrams used during the discussion (Figure 2), dependencies were directly visible. However, in the graphical user interface shown in the evaluation, connections and

[1] For example, one of the experts (Figure 2) commented on the gap between concepts that are meaningful for the system (e.g., the definition of locations A,B,C) and how the user identifies concepts meaningful to them (e.g., how does the user understand the concept of a location in the carpet, is it in the centre or periphery, is in left or right). The mapping of concepts from the mental model to the system model is a programming activity (e.g., defining the borders of locations A,B,C) that remains an open issue in terms of end-user programming.

their rules are not shown. Some way of visualising and inspecting such connections needs to be added.

- Role Expressiveness. This dimension relates to the extent to which users can discern the relation between parts of the program and the whole. As an object can belong to several GadgetWorlds, the effect it has on each is not easy to understand from the physical appearance. Future developments of the GadgetWorld Editor will need a way to illustrate to the user how the specification of the parts influences the dynamic behaviour of the GW (similar to debuggers in Object Oriented environments).

3.2 Evaluation results

Currently e-Gadgets seems to be pitching at an abstraction level appropriate for end-users, but much depends on the quality of the tool support foreseen. The feedback to the project, is captured as four tentative design principles:

- The Gadgetworld behavior 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 GW developers should be supported with at least as good tools as programmers have at their disposal, e.g., 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.

Future sessions using working prototypes are planned to evaluate the ease of GadgetWorld composition and debugging by end-users.

4 Concluding remarks

eGadgets is a bold attempt to provide end-users with the ability to construct or modify ubiquitous computing environments. It takes concepts of component based software development and applies them to treat physical objects as components of Ubiquitous Computing environments. This paper has summarized the concepts of the eGadgets project, the current first experimental demonstrator and our attempts to include considerations of end-user acceptance and usability to the formation of the concepts. Currently, a richer and more robust concept demonstrator is being implemented that will support a test with end-users.

5 Acknowledgements

We would like to thank M.M.Bekker, A.Gritsenko and C.Huijnen for participating in the review.

6 References

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.

A. Kameas, D. Ringas, I. Mavrommati and P. Wason, “eComP: an Architecture that Supports P2P Networking Among Ubiquitous Computing Devices”, in Proceedings of the IEEE P2P 2002 Conference, Linkoping, Sweden, Sept. 2002.