Poster proposal: IT&T 2005, Information Technology & Telecommunications annual conference
Oct 26-27, Cork, Ireland

Scenario-based Methodologies in Identifying Ubicomp Application Sets

Fiona Mahon, Tom Pfeifer, Mícheál Crotty
Telecommunications Software & Systems Group [TSSG]
Waterford Institute of Technology [WIT]
Cork Road, Waterford, Ireland
fmahon@tssg.org, t.pfeifer@computer.org, mcrotty@tssg.org

Abstract: In ubiquitous systems of the future, a requirement for appropriate methods for the selection, analysis and evaluation of applications is evident. Traditional methods of analysis rely heavily on the inherent predictability of the system, something which is totally lacking in ubiquitous systems. This paper discusses one possible approach to flexible analysis and evaluation, the Scenario Based Approach, describing its use in the IST FP6 project, Daidalos. The Scenario Based approach addresses the adaptive nature of ubiquitous systems by its own intrinsic flexibility. This paper explains the approach and argues that it is appropriate that a method of analysis should exhibit similar characteristics to the object of that analysis.

Keywords: Scenario-based design, ubiquitous and pervasive computing, system analysis and evaluation, Daidalos

The aim of ubicomp research is to make computers an invisible, yet integral, part of our everyday lives. Such a paradigm is a major shift from today's compartmentalised structured view of computing systems. This paper discusses metrics and methods for the selection, analysis and evaluation of applications within this new paradigm.

A possible approach to analysis and design of adaptive systems is the *Scenario Based Approach*. This abstract discusses the applicability to ubicomp, from the perspective of a large EU IST FP6 project, Daidalos. In the context of Daidalos, the scenario based approach involves taking a simple life scene that has the potential to use ubiquitous and pervasive systems, and breaks this scene down on many levels and axes. This brings focus and a common thread to what potentially can be an expansive and enmeshed area of research.

Scenarios are composed of a multitude of possible scenes. A single scene forms the basis for an iteration of development. Each iteration includes analysis of the scenario, its implication for development within the Daidalos pervasive environment and the extraction of key applications that need to be developed. Furthermore, it is used at the end of development for external evaluation of the applications.

Daidalos has defined two scenarios, from which all scenes are extracted, and on which all technical discussion, application selection, development and analysis is based. These scenarios are the *University Scenario* and the *Automotive Mobility Scenario*. These scenarios are given loose descriptions, giving scope for definition of many scenes, all which must fit into one of these overall scenarios. The Daidalos System consists of 3 levels of abstraction: a pervasive service platform (PSP), a service provisioning platform (SPP) and the low level access technologies. The PSP contains the following ubiquitous components:

- Context Management;
- Personalisation Management;
- Rules and Event Management;
- Service Discovery and Composition;
- Security and Privacy Management.

The process of delving into a scenario involves creating a Message Sequence Chart (MSC) from a free text passage such as outlined earlier. This activity is very beneficial in initiating discussion on varying aspects of the proposed development. It is especially beneficial in the expansive dimensions of ubiquitous systems, at least allowing persons with varying expertise discuss the same problem from different stances. Since initially the scene was picked to encompass as much innovative technology as possible, a complex set of innovative technology integrations should result. There is also the potential for information exchange between technology fields which may further enhance innovation within project.

Once the MSC has been defined and is agreed between all partners, the lower level details of implementation can be tackled. At this point there are no ambiguities with regards to what should happen within the system. Although the scene remains the driving force of the system, once the interactions of the system are defined, the system design phase is over. Nevertheless implementation continues to focus on the scene, especially on realising it as was originally envisaged.

The next time the scene comes into play is in the evaluation of the system. A novice can read the scene and interact with the system, and it should be clear if the implementation has achieved what it set out to. In this regard the scene acts as a high level test case for the system, pitched at a level easily comprehended by intended system users.

Once a scene has been implemented, the iterative process kicks in. Scenario based development lends itself easily to iterative design. The initial phase is to identify a core scenario with limited scope, and to investigate this confined scenario exhaustively so that all avenues have been explored. Once this phase has been completed, the scenario can be re-examined, expanded and thus allow more and more technologies and opportunities be explored. Iteration occurs by either:

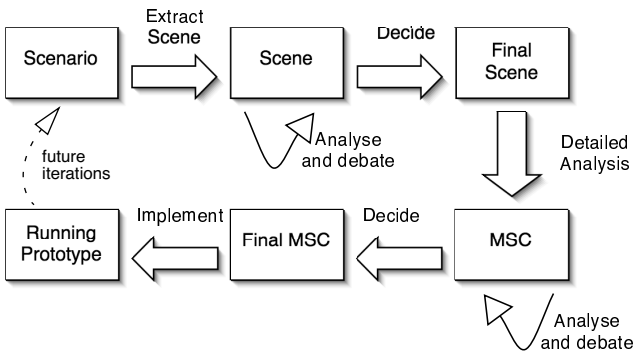

Fig. 1. Flow of the Scenario-based Approach

- Returning to the scenario and defining a new scene (beginning the process again);
- Taking the scene as it is and expanding its scope, possibly enhancing it with new applications.

In this way, the core system becomes more and more generic. The core enabling components become more complete, and eventually it is only the applications at the top level that will need to be added to complete a scenario iteration. Figure 1 illustrates the complete flow within the Scenario-based Approach.

Experience within DAIDALOS has therefore shown that a very simple scene can be used to break down interactions into a very granular level. It begins with a sentence that gives focus to the task at hand. It allows those responsible for components involved to clearly define what is required of them, and to define what they require of other components. It facilitates discussion and results in a clearly defined MSC, and is ultimately the focal point of development and analysis.

Conclusion: The aim of this document was to outline a specific single method of application design and evaluation as used in the Daidalos project, i.e. the Scenario-Based Approach, which could effectively be used in the selection, analysis and evaluation of ubiquitous applications.

The ultimate set of applications would allow a completely ubiquitous scenario, e.g. a University environment that approaches complete pervasiveness. There are many environments that would encounter parallel opportunities and problems to a University, e.g. Government environment or Office environment. Problems and solutions found while creating a ubiquitous University environment are similar, if not the same, as those that would be encountered in these other environments. Daidalos picked two very different scenarios that encompass very different environments and thus increased the potential solution coverage of the project. We believe the Scenario-Based Approach is an adaptive, yet structured way of developing ubiquitous applications. In an environment that itself by definition should be adaptive and invisible, it is important to find a method of design that can encapsulate and account for this. The Scenario Based Approach as used in the Daidalos project has proven to be such a method.

Disclaimer: The work described in this paper is based on results of the EU IST FP6 Integrated Project DAIDALOS. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document reflects only the views of the authors, and is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.