A Framework for Describing Interference in Ubiquitous Computing Environments

Ricardo Morla[1,2] and Nigel Davies[1]
[1]Lancaster University, [2]INESC Porto
ricardo.morla@inescporto.pt and nigel@comp.lancs.ac.uk

Abstract

Future ubiquitous computing environments are likely to consist of numerous interacting components, many of which will have been developed in isolation from each other. Unless appropriate measures are taken, interference (where a component's behavior in a deployed system differs from its behavior when in isolation) is likely to be commonplace. In this paper we explore the importance of this problem and present our work-in-progress on a framework that enables designers, developers, and researchers to describe and thus reason about interference in ubicomp environments.

1. Introduction

Future ubiquitous computing environments are likely to consist of numerous interacting components, many of which will have been developed in isolation from each other. As a result, the exact configuration of components in any given ubicomp environment is unlikely to have been tested and such components liable to interference.

Interference. *A component is said to experience interference when its behavior differs from that which occurs in isolation as a consequence of input from its shared environment being modified, removed, or added.*

We believe that a number of the characteristics of future ubiquitous computing environments are likely to make interference a potential problem. Specifically:

- *Numerous highly interactive components*. We expect that changes in component behavior and, as such, interference will be more likely in environments with more interactive and more numerous components.
- *Diverse set of components*. Given the diversity of components, component design will not likely consider the intricacies of the behavior of other components and thus interference will more likely occur.
- *Isolated development.* We expect the components of future ubicomp environments to have different manufacturers and to be developed in isolation from each other, thus more likely triggering interference.
- *Open environment.* We expect future ubicomp environments to have year- or decade-long lifespans and to be physically open and mostly uncontrolled, thus enabling new components to be brought in that may trigger interference.
- *Unmanaged environment.* We expect future ubicomp environments to be largely unmanaged, allowing for unresolved failures to trigger additional interference.

Given these foreseen characteristics we expect that, unless appropriate measures are taken, interference between the components of future ubicomp environments will be commonplace.

Moreover, we expect interference to be commonplace throughout the many different layers of a ubicomp system – ranging from sensing and actuating of the physical environment and low-level communications through to distributed software and high-level user interaction. Hence, interference will be a concern for all ubicomp developers.

Despite the importance of this problem (as highlighted also in e.g. [1], [2]) there has been no significant research into interference in ubicomp environments. In particular, there has been no research into helping people reason about interference throughout the layers of a ubicomp system (e.g. physical, device, network, middleware, application, and user). The key contribution of this paper is a high-level framework that enables its users to describe and reason about interference in ubicomp environments. In essence we aim to provide ubicomp researchers with a common frame of reference when addressing problems of interference. We believe that this framework

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

provides an important first step towards enabling a systematic treatment of the problem of interference in ubicomp environments.

We have chosen to explicitly target our framework at ubicomp researchers, developers, and designers. We hope our framework can provide its users with benefits broadly similar to those provided by other frameworks throughout computer science. For example, in the field of CSCW, frameworks such as that proposed in [3] have subsequently proven to be useful in the analysis of CSCW systems, in classifying, partitioning, and distinguishing different CSCW research fields, and as a design tool.

Our framework is concerned with the interactions between ubicomp components and their environment. It does not provide a model of the internal behavior of components or of their environment. As such we cannot expect it to be used to automatically identify interference – although future tools that are developed to automatically detect and identify interference may use our framework for describing such interference.

2. Interference Framework

At the core of our framework is a model based on the definition of interference in section 1. This model has three elements: the two interfering components (which we term *source* and *subject*) and the *medium* through which the two components interfere (see fig. 1). Our model defines the following events that involve these elements and that lead to interference. **1**. One of the elements moves to a state in which it can generate an output to the environment. **2**. This element generates an output that is received by the environment. **3**. The environment changes due to the output of this element. **4**. This change causes the input from the environment to the other element to be different from that when in isolation. **5**. The difference in input triggers a change in behavior compared with that when in isolation, and thus interference.

For a source to interfere with a subject, these five events must be causally related. However, as we do not model the behavior of source, medium, or subject we cannot claim that, for example, a change in the environment (**3**) always causes the input of the subject to change (**4**).

Note that the same ubicomp components may be involved in multiple instances of interference (e.g. the source has two different outputs to the same medium, both of which cause interference). Therefore, to uniquely identify an instance of interference users of the framework must identify source, medium, and subject as well as the details of their interactions and behavior. Moreover, note that our framework has been developed with the intent of supporting the description of interference at all levels of a ubicomp system. The concepts of source, medium, and subject and the interactions between these elements have no applicability restrictions.

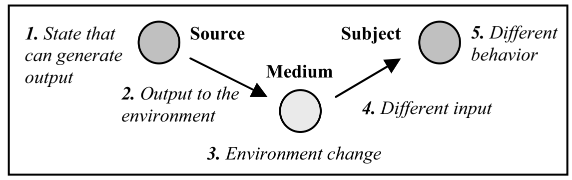

Figure 1. Model of interference of the proposed framework

2.1. Notations for Describing Interference

We propose graphical and textual notations for describing interference in ubicomp systems. Both notations can be used to define the topology of the system and to describe the instance of interference that occurs in that system. The topology includes the components that are part of the system and their interactions or dependencies. Users can map the causally related events that identify the instance of interference to the topology of the system. Note that as the framework does not model component behavior, our notations are not restricted to any behavior description.

The graphical notation that we propose is based on a directed graph similar to that of fig. 1. The nodes and edges of the graph represent the topology of the system. A set of labels can be affixed to the nodes and edges of the graph to identify the instance of interference (i.e. to describe the behavior of source, medium, and subject and their interactions).

The textual notation that we propose is based on tuples. We use standard approaches to represent the directed graph of the system topology in textual form using nodes and edges. An n-tuple represents the n nodes of the graph and a set of ordered pairs of nodes (2-tuples) represents the edges. The instance of interference can be described using a 5-tuple. Each field of this 5-tuple is a 2-tuple that represents component behavior or component interaction and the location in the topology of the system to which such behavior or interaction applies (i.e. source, medium, or subject nodes, and source-medium or medium-subject edges).

By using these notations to describe interference we expect the framework user to be able to communicate to others that the behavior of these components and their interactions are or may potentially be causally related and involved in an instance of interference.

2.2. Classes of Interference

We identify three distinct types of interference based on how the changes in the medium cause the input to the subject to differ from that observed when in isolation.

Generative Interference. Changes in the medium generate additional input to the subject. This typically causes the subject to behave as if it were reacting to an event that did not occur. **Destructive Interference.** Changes in the medium cause input to the subject to be removed. This causes the subject to behave as if it were failing to react to events. **Distortional Interference.** Input from the source is combined with other input to the medium such that it modifies the input to the subject. This causes the subject to behave as if it were reacting to a different event.

We expect that, in the future, it may be possible to ascertain the types of interference solution that can address an instance of interference (for example the types presented in the next section) from the type of interference instance.

2.3. Stock Solutions to Interference

We present five generic solutions that can address any interference instance described with our model. This set is complete, i.e. given any interference instance that can be described using our framework, all possible solutions to such interference will in practice be instances of one or more of these generic solutions.

Our first three generic solutions modify or remove the source, medium, or subject such that 1) the source does not generate output that modifies the medium and causes interference; 2) input from the source either does not trigger changes in the medium or these changes are not output to the subject; and 3) a medium input different from isolation does not trigger a similarly different subject behavior, and hence interference. These solutions might take a number of forms, e.g. redesigning components or coordination with the other components potentially involved in interference.

Modifying or removing a component might not always be possible if this component is e.g. a third-party component or part of the physical environment. The other two generic solutions filter source-medium and medium-subject interactions without changing these components. A source-medium solution prevents the part of the source's output that would cause interference from changing the medium, while a medium-subject solution filters the output of the medium and prevents this output from triggering affected behavior in the subject.

3. Evaluation

The evaluation of conceptual tools such as the framework described in this paper is a challenging task – appropriate quantitative measurements are difficult to obtain and the usefulness of other frameworks has typically been assessed only through extensive use. In fact, we expect that the benefits of the framework will only be evident on average and over the years and numerous ubicomp systems – namely, we expect that on average the framework users will be able to more quickly rule out the components of their systems that are not involved in interference; and that on average framework users will consider more solutions to interference in their systems and thus be able to develop better solutions to such interference. Furthermore, there are at the moment no deployed ubicomp systems in which interference is likely to be a serious issue and with which we could evaluate our framework.

Despite the obvious difficulty of evaluation, we have sought evidence of the utility of our framework in the form of a series of examples of its use. These examples illustrate four distinct applications of the framework: as a means to precisely describe a wide range of instances of interference; as a basis for an interference-focused analysis of a ubicomp system and for deriving a methodology for such analysis; as a basis for an analysis of a potential solution to interference; and as a means to express a process for detecting interference. Due to space limitations we do not provide more details about these examples here.

We are planning to run a user trial to gain a better understanding of how people actually use the framework to address interference in a system. We do not expect to single out the framework as the factor that enables people to solve interference, but nonetheless plan to have different control groups from which we can compare results and learn how the framework is used.

4. Related Work

Research on feature interaction addresses the issue of separate software entities sharing a resource [4] and thus potentially interfering with each other. Extensive research on formal methods, online approaches, and software engineering techniques has been conducted with the aim of addressing this problem. In contrast to work on feature interaction, our work supports only the description of interference between components but targets interference at all levels of a ubicomp system (c.f. focusing on interacting software entities).

The work in [2] provides examples of interference in ubiquitous computing. However, the focus of [2] is on user understanding of the ubicomp system and not on the problem of interference and on reasoning about this problem.

The work in [5] was a source of inspiration for our work. Although the work in [5] provides a solution to prevent interference, it does not consider all the layers of ubicomp (e.g. user layer). Furthermore, it defines a taxonomy of interactions that cannot be used to disambiguate different instances of interference (namely MAI and STI interactions [5]) or to describe distortional interference.

System modeling approaches provide simple constructs that can be used to describe the behavior of the elements of complex systems (e.g. [6]). Although our framework does not model component behavior, it is likely to be usable in tandem with system modeling tools or with formal methods [4] to automatically detect interference.

We recognize that interference and associated solutions are not restricted to the technical issues we have discussed in this paper. For example, we expect legal, social, and economic aspects to be important in reasoning about interference in the deployment of ubicomp systems. Interference may occur between different authority domains an as such be socially inappropriate, illegal or make the ubiquitous computing environment economically unviable. Moreover, non-technical solutions such as branding and compliance testing are likely to be important aspects of general solutions to the problem of interference.

5. Conclusions

Without research into appropriate counter-measures interference is set to be a significant problem in future ubiquitous computing environments. This paper provides a framework that can be used to describe and reason about this problem.

We expect our framework to provide the basis for future research into interference in ubiquitous computing in areas such as automated interference detection and resolution, potentially using plug-ins that could allow, for example, source-medium filtering to be effected at the source or at the medium without the need to modify these components – downloading adequate filter plug-ins for potential sources and media.

Finally, we observe that our framework is proving to be fast to take up and useful in daily discussions within our group. As we tackle new ubicomp deployments those researchers who have experience with our framework share a common vocabulary and are able to express ideas and concerns relating to interference quickly, precisely and unambiguously. This first-hand experience of the benefits of having a framework for discussing interference gives us confidence that we have a useful tool in the drive towards interference-free ubiquitous computing environments.

References

[1] N. Davies and H.-W. Gellersen. “Beyond prototypes: challenges in deploying ubiquitous systems.” *IEEE Pervasive Computing*, 1(1):26-35, Jan. 2002.

[2] K. Edwards and R. Grinter. “At Home with Ubiquitous Computing: Seven Challenges”. In *Proceedings of Ubicomp 2001*, pages 256-272, Sep. 2001.

[3] A. Dix. “Computer-Supported Cooperative Work – A Framework”. Chapter in *Design Issues in CSCW*, pages 23-27. Springer-Verlag, 1994.

[4] L. Blair, G. Blair, J. Pang, and C. Efstratiou. “Feature Interactions Outside a Telecom Domain”. In *ECOOP 2001 “Feature Interaction in Composed Systems” Workshop*, June 2001.

[5] M. Kolberg, E. H. Magill, and M. Wilson. “Compatibility Issues Between Services Supporting Networked Appliances”. *IEEE Communications Magazine*, 41(11):136-147, Nov. 2003.

[6] G. Bellinger. “Introduction to Systems Thinking”. In *http://www.systems-thinking.org/intst/int.htm,* Dec. 2005.