Why is Context-Aware Computing Less Successful?

Waltenegus Dargie
Chair of Computer Networks
TU Dresden
Dresden, Germany
waltenegus.dargie@tu-dresden.de

ABSTRACT

The paper addresses some of the challenges in developing context-aware systems. It begins by looking into some of the proposed context-aware systems and examines why they fail to have practical relevance. It then examines some of the successful features of contemporary ubiquitous systems and applications to drive a lesson that can be useful for developing context-aware systems.

ACM Classification Keywords

C.2.4 Computer-Communication Networks: Distributed Systemsdistributed applications; H.4 Information Systems Applications: Miscellaneous; H.5.3 Information Interfaces and presentation: Asynchronous interaction—*collaborative computing, theory and models, synchronous interaction*

General Terms

Algorithms, Design, Experimentation, Management, Measurement, Performance, Reliability, Verification

Author Keywords

Context-aware computing, pervasive computing, ubiquitous computing

INTRODUCTION

More than 20 years have passed since Mark Weiser first articulated his vision of ubiquitous computing. In his seminal work, Weiser argues that the computers of the 21st century should be able to "weave themselves into the fabric of everyday life" until they become indistinguishable from it. Weiser identifies several manifestations of the coming into fruition of this vision – computers will become available in different sizes and capacity, the use of wireless communication will become pervasive, displays that fit various contexts will be designed, and, computers will become context-aware.

Following this, context-aware computing received a significant amount of attention. In the beginning, much of the focus has been on developing indoor localisation techniques that can capture the location of persons and objects. Infra-red, RF, and ultrasonic signals were extensively studied to achieve high estimation accuracy. Soon the investigation extends itself to the recognition of other type of contexts, such as the activity and identity of users, devices, and physical environments. Electronic calendar as well as accelerometer, temperature, humidity, and light sensors were used to capture some of the aspects of these contexts. Moreover, models and architectures were proposed to transform raw sensor data into meaningful concepts or abstractions of real-world settings and to enable the development of context-aware systems and applications [1].

Through the years, however, it has not been possible to employ context-awareness in practical systems and applications. In contrast, most of the other aspects of ubiquitous computing have become realities: we now have computers of diverse size and capacity and we can carry them with us easily; wireless technology has become both affordable and pervasive; and information displays are available in various forms and sizes. Whereas giving a complete and satisfying explanation for the lack of success may not be feasible, it is helpful to examine some of the problems associated with the design, development, and deployment of context-aware systems.

The first context-aware systems were the active badges [2]. These systems recognise a user's whereabouts and activities (in an indoor environment) and redirect incoming calls through a fixed line telephone network to the nearest location. These systems were soon outdated by mobile phones which can be carried by users everywhere. The next development was to make the mobile phones themselves context-aware by integrating several Microelectromechanical Systems (MEMS) sensors [3]. The idea was to capture the social and conceptual context of a user and to handle incoming calls accordingly. For example, calls can be admitted, rejected, or redirected; the status of the mobile phone can be dynamically changed to active, silent, or vibration mode.

This idea was not successful because of two main reasons: (1) the "miss-call" service on most mobile phones proved to be a useful, simple, and asynchronous solu-

tion for controlling incoming calls (it gives users the autonomy as well as the passivity of admitting, delaying, or altogether rejecting a call); and (2) capturing simple conceptual settings such as "being in a meeting" or "driving a car" was not as simple as it was initially thought. The problem with the second reason was not only the technical difficulty of signal processing but also the complexity of determining when, for how long, and at what frequency a user's context (activity) should be sensed or computed.

As time went on, the idea of capturing the context of everyday objects encompassed to integrating sensors into headphones [4], coffee mugs [5], and homes [6, 7], among other things. Moreover, additional context sources, both direct and indirect sources, were employed to enrich the knowledge of recognition tools [8]. Remarkable accuracies in context-recognition were reported to have been achieved. Regardless of these achievements, however, still real-world applications are missing. So it is natural to ask whether context-aware computing is really relevant.

Davies and Gellersen investigate (and the topic was revisited by Davies ten years latter [9]) some of the technical, social, legal, and economical challenges facing the deployment of ubiquitous systems [10]. Though these challenges apply to context-aware systems as well, it is also important to look into some of the aspects of Weiser's vision itself. To begin with, while Weiser asserts that the ubiquitous availability of computers will (1) liberate users from becoming computer addicts and (2) help overcome the problem of information overload, a closer look into the scenario he proposes reveals that ubiquitous deployment of computers will, in fact, necessarily make human beings dependent on them, however imperceptible this dependency may be.

Therefore, one of the technical difficulties in realising the vision is enabling context-aware systems to continuously sense and capture the context of a user[1] and to accurately distinguish the transitions between different contexts, each of which may require a different domain of knowledge to be correctly interpreted. This will, of course, necessitates the processing of a vast amount of data and the management of these data in a meaningful way. Both tasks are practically overwhelming. Secondly, the scenario gives little attention to the sense of purpose and accomplishment human beings drive (and the security and confidence that come with them) when they engage themselves in solving some real-world problems. Human interaction with humans as well as their surrounding surpasses the mere need of solving an immediate problem; it has complex and often intrinsic existential values. Which means, humans will and should solve some problems on their own. In this sense, it is difficult to quantify at any given time which real-world

problems are worthy of human efforts and which of them should be left to computers.

Thirdly, the scenario does not properly examine the concomitant effects and side-effects of involving computers in everyday human decisions. Weiser himself points out that privacy will be the most conspicuous issues with ubiquitous computing. For example, as Davies and Gellersen correctly point out, the same system that helps users find a lost key or purse in a parking place can also be misused to harm people or violate human privacy. This partly explains why we do not find in real life intelligent smart places [11] and chat applications [12].

LESSONS FROM CONTEMPORARY SYSTEMS

Unlike Weiser's vision of ubiquitous computing, which literally pushes computers into the background but leaves them to be there at all times, a cursory examination of the most successful contemporary applications and systems reveals that their success is partly due to their reactive nature. They can be left aside or turned off at any given time. Moreover, while providing a highway to a vast amount of data, these systems enable their users to make decisions pertaining to the type, size, and duration of data that can be admitted or made available. The applications may provide recommendations or a short list of competing contents based on their knowledge of the users' preference or current "situation", but essentially, the users make the final decision. The author is not alluding, of course, that these systems cannot be misused or that there is little concern for privacy or that they are not addictive or that they do not overload users with information. The focus is made simply on their overt reactive nature[2].

The understanding of a context of use has been successful in other areas as well, such as compiler design and speech recognition, in which close-world models are strictly employed to make intelligent decisions. These models assume that the context of a setting, an interaction, or an environment can be known with complete certainty or a "best" definite answer can be derived from incomplete information. By clearly defining the scope and usefulness of these models, people have developed practical systems and simplify the difficulty of computing.

So what lessons should designers of context-aware systems learn? The first lesson is that reactive systems that provide unobtrusive services when asked are preferred by most users than proactive systems. Secondly, a context-aware system should have a clearly defined operation context. In other words, the beginning and end of the task of the system should be clearly known.

[1]Here, the term context is used broadly to refer to the state of the user, the devices with which the user interacts, the physical surrounding, etc.

[2]Murad Ahmed of The Times (The Sunday Times) writes a column on how Google's Ad Services can manipulate rich context information to overwhelm users with tailor made advertisements:
(http://technology.timesonline.co.uk, March 11, 2009).
Similar concerns in the area of healthcare are raised in [13].

This not only limits the amount of data that should be gathered and processed, but also improves the quality of the contexts that should be detected (captured). Third, the number of distinct contextual states (context values) should be limited. This, as a result, limits the operational scope of the system and makes it predictable. This aspect has indeed several advantages:

1. Analysing the potential side-effects of the system will become tractable;

2. The system can be comprehensible during troubleshooting;

3. The time to recognise a context of interest becomes short (this is essential for real-time systems); and,

4. The system can be configured by average users.

The more sober lesson we should learn is, however, that the success of context-aware computing really depends on the existence of noble and practical systems that react upon the occurrence of a piece of context. It is one thing to integrate some form of intelligence into everyday objects such as headphones, coffee cups, or furnitures, but another to know what to do with them. The latter aspect has't got as much attention in the past as the former. Maybe it is now time to equally divide the focus between the two aspect.

REFERENCES

1. W. Dargie, "The role of probabilistic schemes in multisensor context-awareness," in *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, ser. PERCOMW '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 27–32. [Online]. Available: http://dx.doi.org/10.1109/PERCOMW.2007.115

2. R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, pp. 91–102, January 1992. [Online]. Available: http://doi.acm.org/10.1145/128756.128759

3. D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong, "Sensay: A context-aware mobile phone," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ser. ISWC '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 248–. [Online]. Available: http://portal.acm.org/citation.cfm?id=946249.946884

4. P. K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *BSN*, 2009, pp. 144–148.

5. H. W. Gellersen, A. Schmidt, and M. Beigl, "Multi-sensor context-awareness in mobile devices and smart artifacts," *Mob. Netw. Appl.*, vol. 7, pp. 341–351, October 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1016587515822

6. S. Intille, K. Larson, E. Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," *Pervasive Computing*, vol. 3968, pp. 349–365, 2006.

7. K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darrell, "Activity zones for context-aware computing," in *UbiComp 2003*. Springer-Verlag, 2003, pp. 90–106.

8. W. Dargie, "Adaptive audio-based context recognition," *Trans. Sys. Man Cyber. Part A*, vol. 39, pp. 715–725, July 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1656581.1656584

9. N. Davies, "Beyond prototypes, again," *IEEE Pervasive Computing*, vol. 10, pp. 2–3, January 2011. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2011.2

10. N. Davies and H.-W. Gellersen, "Beyond prototypes: Challenges in deploying ubiquitous systems," *IEEE Pervasive Computing*, vol. 1, pp. 26–35, January 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=612822.612830

11. H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *Knowl. Eng. Rev.*, vol. 18, pp. 197–207, September 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=991804.991806

12. A. Ranganathan, R. H. Campbell, A. Ravi, and A. Mahajan, "Conchat: A context-aware chat program," *IEEE Pervasive Computing*, vol. 1, pp. 51–57, July 2002. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2002.1037722

13. M. A. Winker, A. Flanagin, B. Chi-Lum, J. White, K. Andrews, R. L. Kennett, C. D. DeAngelis, and R. A. Musacchio, "Guidelines for medical and health information sites on the internet: principles governing ama web sites," *American Medical Association. JAMA.*, vol. 283, 2000.