Using Context as a Crystal Ball: Rewards and Pitfalls

Keith Cheverst, Nigel Davies, Keith Mitchell and Christos Efstratiou

Distributed Multimedia Research Group, Department of Computing, Lancaster University, Lancaster, UK

Abstract: Context-awareness can be used to simplify a user's understanding of, and interaction with, interactive systems. In effect, through adaptation, context-aware systems can migrate complexity away from the user and into the system (or agent). However, the incorporation of context-awareness raises a number of issues. For example, users are required to trust the behaviour of the system's intelligence and this requires the system to have predictable behaviour and the ability to successfully and consistently preempt the user's goal. Unfortunately, the agent may incorrectly preempt the user's goal, owing to either flawed intelligence or to incorrect or out-of-date contextual information. In such circumstances the user is likely to feel frustration because the system will either appear overly prescriptive or, worse still, present incorrect results. This paper considers these issues, a number of which are described in anecdotal form, based on our experiences in developing and evaluating the context-aware GUIDE system.

Keywords: Context-awareness; Design; Interactive systems; Usability

1. Introduction

The use of context (both situated and environmental) has significant potential for simplifying a user's understanding of, and interaction with, complex interactive systems. This potential is highly valuable given the continuing consumer demand for greater levels of functionality from highly mobile devices, e.g. WAP phones. The interactive systems that run on such devices are likely to share very limited input/output bandwidth at the interface between the system and the user. Consequently, techniques for simplifying patterns of interaction are both desirable and necessary in order to enable such devices to have the ease of use associated with the 'Information appliance' [1].

This paper discusses some of the potential rewards and pitfalls that can await designers wishing to incorporate context-awareness [2] into interactive systems. Many of the issues are described in anecdotal form, based on our experiences gained through the development and evaluation of the context-aware GUIDE system [3].

In order to introduce some of the issues that arise, consider that popular interactive system the car and, more specifically, its braking system. The Antilock Braking System (ABS) is a context-aware system that was introduced as a safety measure to reduce a car's braking distance and diminish the potential for a driver to cause their car to skid through excessive braking. The context sensed by the antilock braking system includes:

a) whether the driver is currently trying to brake (i.e. situated context),
b) whether or not the wheel is currently "locked" under braking (i.e. environmental context).

The adaptive element of the system involves detecting when the wheel is locked and then decreasing braking force until the wheel is no longer locked. Once the wheel is no longer locked (and the situated context is still that of braking) further braking force is applied to the wheel.

Before the advent of ABS, drivers were required to develop mental models that took into account the complex interrelationship that exists between braking force and the friction between the car's tyres and the road.

Consideration of the ABS system allows us to identify an agent, acting on behalf of the driver, which reduces the mental and physical demands of driving the car. In effect, the agent takes some control (or power) away from the driver and (providing the driver prefers less rather than

Personal and Ubiquitous Computing (2001) 5:8–11

more interaction with the car) makes the car easier to drive.

In effect, the ABS system enables the driver to form a simplified mental model regarding the car's braking system, i.e. drivers don't need to have such a detailed comprehension of the rules governing "excessive" braking force. However, if the car's driver is used to a conventional, manual braking mechanism they might have learnt the skill of "pumping the brakes" in order to prevent the car from skidding. Unfortunately, if this skill is employed by the driver of a car with ABS, the two approaches can conflict causing the braking distance to be increased. This example highlights three potential pitfalls that can arise from adapting to context, namely:

1. The problem of failing to reach a stable state [4]. If both the user and the system attempt to adapt to the current context then it is unlikely that the system will manage to reach a stable state. Under such circumstances, the system is likely to appear unpredictable. When designing context-aware systems, it is clearly important to consider the background/expertise of the user, i.e. are they likely to have already formed a mental model for interacting with a similar (non-adaptive) system?
2. The trade-off between prescription and freedom. If the driver desired, for whatever reason, to lock the wheels of the car then the system would prevent them from achieving this task.
3. The user must trust the agent performing the adaptation. When ABS was first introduced, there was, not altogether surprisingly, some mistrust of the system by drivers. Indeed, the driver who comprehends the workings of the ABS system is required to trust both the context-sensing technology and the intelligence of the agent, i.e. its infallible ability to react appropriately to the context in a failsafe manner.

Building upon the work of Schmidt [5], it is possible to identify three main ways in which context can be used to simplify the user's interaction with an interactive system:

1. Simplifying/reducing the task specification required from the user in order to achieve their desired goals, i.e. reducing the need for input/action by the user. At one level this can simply mean filling in a required blank, such as the user's current location, based on information that is sensed by the system. However, at a higher level, it can also involve attempting to preempt the user's current goal (e.g. quickly bringing the car to a halt without locking the car's wheels) in order to reduce their task specification.
2. Changing the output produced by the system, i.e. reducing the quantity of information that has to be processed by the user or increasing the quality of information presented. Once again, some reduction in output might be achievable by attempting to preempt what output is likely to be required by the user.
3. Reducing the complexity of rules constituting the user's mental model of the system. This is generally achieved by some form of intelligent agent that performs some portion of the required computation on the user's behalf.

The following section describes and analyses some of the positive and negative experiences of using context gained through our development and evaluation of the GUIDE system.

2. Experiences of Developing a Context-Aware Tourist Guide

The GUIDE system has been developed to provide visitors to the city of Lancaster with information that is tailored to the current context. The city contains a number of strategically-positioned wireless communication cells with a diameter of approximately 300m depending on the layout of buildings. These communication cells are used for disseminating location information and tourist information to mobile GUIDE units. By carrying a GUIDE unit, visitors can receive up-to-date information about the city's attractions while following a structured tour of the city tailored to their specific requirements.

The following sub-sections describe the way in which GUIDE adapts to context.

2.1. Simplifying/reducing the task specification

One type of context utilised by GUIDE is that of the visitor's location and the location of attractions within the city. The system reduces the need for input by assuming that the information required by the visitor is strongly influenced by their current location. So, for example, a visitor standing outside Lancaster Castle can request the

system to "Tell me about the area I am in" as opposed to searching the contents page for "Information on Lancaster Castle". However, an initial version of GUIDE made the mistake of allowing visitors only to obtain information regarding their current location. In this case, the over-determination [4] employed by the system to simplify the visitor's task was inappropriate. In more detail, the lack of flexibility in this version frustrated users and so the system was extended to enable visitors to specify their requirements more fully, e.g. by searching for information using a keyword. This anecdote helps to illustrate the difficulty of successfully (and consistently) preempting the goal of the user.

2.2. Changing the output produced by the system

In general, the GUIDE system attempts to constrain and tailor information presented to the visitor based on both personal and environmental context. So, for example, when the visitor requests a list of nearby attractions, the list is constrained in such a way that those attractions that are open, and have not already been visited, are placed higher up the list. In this case, the assumption is made that the visitor is more likely to be interested in attractions that are open and that have not already been visited. An earlier version of the system constrained the output by removing all closed attractions from the presented list. However, this frustrated some visitors who were interested in visiting the attraction anyway, e.g. to view the architecture of a building. This further demonstrates the difficulty of preempting the user's goal. A future version of the system will use the visitor's stated interest in architecture to determine whether certain closed attractions are included in the list.

While evaluating the GUIDE system, we experienced some difficulty capturing the visitor's location context with sufficient accuracy. The likely consequence of obtaining inaccurate or incorrect contextual information is that the adaptation performed by the system will, in turn, be inappropriate. In the case of GUIDE, when presenting a list of "nearby attractions" to the visitor, some of the attractions were not always as "nearby" as might have been expected.

2.3. Reducing the complexity of the user's mental model

In GUIDE, the agent that acts on behalf of the user is designed to relieve the user of the onerous task of studying maps and guidebooks in order to devise and follow an interesting tour. In more detail, the agent calculates tours based on a variety of different contexts, such as the visitor's current location, the current time, special opening hours of attractions, the relative positioning of attractions in the city and the preferences of the visitor, e.g. an interest in historic buildings.

3. Strategies for Building Context-Aware Applications

A number of strategies can be identified for the design of interactive systems that utilise situated and/or environmental context. The following strategies are based on our analysis of existing context-aware interactive systems, such as ABS, and the issues that were experienced during the development and evaluation of the GUIDE system:

1. When using context to constrain the presentation of information, or to simplify the specification of a task, it is crucial that the adaptation does not inappropriately over-determine the user's interaction. This issue is basically a specialisation of the fundamental design trade-off between prescription and freedom/flexibility.

2. When considering adaptation to context, designers should be careful to bear in mind the principle of least astonishment and the need for predictability. If designed well, then adaptation to context has the potential to increase the integral predictability/consistency of the system. However, as described in the ABS example, the inappropriate transfer of skills can cause difficulties.

From a more technical perspective, the following issues need to be considered when engineering context-aware systems.

a) The sensing technology used for obtaining context needs to be dependable. This means both accurate and available in a timely manner.
b) The intelligence of the agent responsible for adapting to context needs to be sufficiently

flexible to enable users to override the adaptation strategy.

4. Conclusion

This paper has considered some of the potential rewards and pitfalls of utilising context in the design of interactive systems. On the positive side, adaptation to context can be used to simplify a user's understanding of, and interaction with, interactive systems by migrating complexity away from the user to some form of intelligent agent. This agent must effectively preempt the user's goal (based on the current context) and this is where difficulties can arise. For example, the agent could erroneously presume the user's goal, either because of flawed intelligence or because of incorrect or out-of-date contextual information.

References

1. Norman D. The invisible computer. Cambridge University Press, 1999
2. Schilit B, Adams N, Want R. Context-aware computing applications. In: Proceedings of the Workshop on Mobile Computing Systems and Applications. Santa Cruz, CA, 1994;85–90
3. Cheverst K, Davies N, Mitchell K, et al. Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of CHI'00. ACM Press, New York, 2000; 17–24
4. Thimbleby H. User interface design. ACM Press, New York, 1990
5. Schmidt A. Implicit human-computer interaction through context. In: Brewster S, Dunlop M (eds) Proceedings of the 2nd Workshop on Human Computer Interaction with Mobile Devices. Edinburgh, Scotland, 1999; 23–27

Correspondence to: Keith Cheverst, Distributed Multimedia Research Group, Department of Computing, Lancaster University, Lancaster LA14YR, UK. Email: {kc, nigel, mitchelk, efstrati}@comp.lancs.ac.uk