Investigating Context-aware Information Push vs. Information Pull to Tourists

KEITH CHEVERST, KEITH MITCHELL AND NIGEL DAVIES

Distributed Multimedia Research Group

Department of Computing

Lancaster University

Lancaster, LA14YR, U.K.

Tel: +44 (0)1524 594539

Fax: +44 (0)1524 593608

E-mail: {kc, km, nigel}@comp.lancs.ac.uk

Abstract

Despite much interest over recent years in the area of context-aware computing, there are still a number of significant gaps in our understanding of the HCI issues associated with such systems. One particular issue that remains relatively unexplored is how to design around the apparently conflicting goals of adaptating to changes in context while at the same time adhering to the principle of predictability. In this paper, we describe our investigations into this issue through two alternative designs of an interactive context-aware tourist guide. One design is based around information pull, i.e. the emphasis is on the user to decide when context-aware information is presented. The other design is based on information push whereby the actual presentation of context-aware information is triggered by contextual events, e.g. changes in the user's location or changes to the opening times of attractions. Through the evaluation of these alternative designs we hope to gain a better understanding of the usability implications of push vs. pull in both this specific domain and in interactive context-aware systems in general.

1. Introduction

One of the fundamental attributes of context-aware systems [1,2] is that they react to not only the user's input but also input (i.e. context) from the user's environment. For example, the classic location-aware system utilises location in order to tailor the information presented to the user. The inherent problem with any such system is that designers need to carefully balance the way in which the system reacts to environmental triggers (e.g. location) with a desire for the system

to behave in a predictable manner and adhere to the principle of least astonishment [3].

Consider, for example, the classic interactive conext-aware application, namely, the location-aware visitor guide (e.g. [4]). There are two key alterntive approaches that such a system can employ when the user changes location. The first approach would be to push the location-aware content to the user in order to reflect his or her new location. This approach could surprise the user with the timing of the presentation or possibly overwrite the content that the user was currently engaged in reading. The second approach would be to wait for the user to pull the location-aware content. The implication with this approach is that the the system would leave the currently displayed content unchanged, despite the fact that the content has become inconsistent with the user's current location. Another implication of this latter approach is that the designer needs to consider what kind of user notification mechanism may be required in order to prompt the user to pull the new content.

The problem of desiging systems that react to context is compounded when one considers that location may not be the only dynamically changing contextual trigger that may occurr. For example, if the state of a tourist attraction (i.e. whether it is currently open or closed) is also used as context then what action should be taken when this piece of context changes?

In this paper, we use the GUIDE system [5,6] as a research vehicle to explore these issues. As part of this exploration, we will utilise field-trial studies in order to investigate and study the reaction of city visitors to the two fundamentally different approaches, i.e. information push and information pull.

2. Background

When designing any technology to support a tourist's exploration of a city, one needs to recognise that the primary aim of the technolgy should be to assist the visitor in experiencing the real city. Clealy, the technology should not consume too much of the user's attention. An initial evaluation of the GUIDE system (documented in [6]) revealed that visitors generally enjoyed using the system to explore the city and did not focus too much of their attention on interacting with the system itself. However, one of the weaknesses of the initial system and its subsequent evaluation, is that it did not consider how the presentation of dynamic

information, such as changes to opening times etc, could be communicated to the user in an appropriate way.

2.1 GUIDE's Pull-Based Approach

The original GUIDE system only considered one approach for managing the presentation of context-aware information. In more detail, the approach was ostensibly based on information pull whereby a user initiates the presentation of context-aware information by, for example, tapping on the information button. This pull-based approach was adopted for a number of reasons. Firstly, the approach seemed the safest way to prevent visitors spending too much time focussed on the GUIDE display, i.e. poised waiting for new information to appear. Secondly, the approach seemed to match the kind of interaction that takes place when using a guidebook for the task of exploring and learning about a city. In this case the visitor retrieves (or pulls) information for a given attraction by turning to the appropriate page in the guide. In effect, our system used context to simplify the retrieval task by using the visitor's current location in order to pre-empt the appropriate page of information. However, the guidebook approach has problems when one considers the handling of dynamic information because the user expects the information to remain constant.

The pull-based approach also fitted well with a standard web-based browsing metaphor/interaction model. In order to adhere to this metaphor a refresh button was incorporated in the system's GUI. When roaming into a new area, users were required to press this button in order to cause the system to display information appertaining to their new location. However, the field-trial revealed that many visitors were unsure of when the button should be pressed.

In addition to using the system as a guidebook, a tour-guide mode was also supported. In this mode of operation, the system would guide the visitor on a structured tour by presenting the visitor with a series of instructions assisting him or her to navigate from their current attraction to the next attraction in the tour. However, once the visitor arrived at the next attraction in the tour, the system did not demonstrate any proactive behaviour. Instead, the system would effectively fallback to guidebook mode and require the user to tap the information button in order to retrieve information pertaining to their current location.

The only real exception to the original system's pull-based approach was the incorporation of a visual notification window which would attempt to notify the user of their current location (as shown below in figure 1).

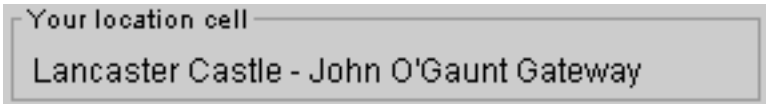

Figure 1: The visual notification window.

However, the cell-based approach for sensing location meant that the accuracy of the information presented was often approximate and therefore of limited utility to the visitor.

2.2 Analysis

Analysing the role of information pull in the original system identified the following two key points.

- The asynchronous nature of information pull meant that the information currently displayed to a user could become inconsistent with his or her current context. For example, if the information currently displayed on the GUIDE unit's screen states that the user is standing next to the castle but the visitor has moved from the castle to the priory, then, the presented information has effectively become inconsistent with the current context, i.e. location. Furthermore, a change in context may relate to context other than location, e.g. a change to the opening time of an attraction. Communicating such context changes were not investigated in the original version of the system but clearly raise many interesting issues that demand investigation.
- The stimulus used to cause the presentation of new information to occur comes directly from the user and therefore the user is already focussed on the system when the new content is presented. Consequently, the user's current task, e.g. enjoying a view, is not interrupted or disturbed by the presentation of context-aware information. However, clearly some tourists do choose to follow tour guides who effectively push (albeit verbally) information to the user based on the current location. It is therefore a shortcoming of the original system that this approach was not explored. In particular, it is important to explore the usability issues associated with information push vs. information pull and the additional infrastructure requirements required for supporting information push. For example, to what extent is the cell-based location

granularity incorporated in GUIDE suitable for providing triggers that are sufficiently accurate to act as stimuli for information push.

3. Exploring Information Push

In order to investigate the usability issues associated with information push, we have developed a push-based version of the GUIDE system. This version is about to be evaluated via a period of field-trial study (tourists tend to visit Lancaster in the summer months). In particular, we aim to investigate the extent to which the possible benefits of information push, such as greater spontenaety and reduced effort in retrieving information, compare with the likely negative aspects such as interruption, reduction in user control, and so forth.

3.1 Implications

One of the key implications of following a strict information push route is that by pushing new content to the user, the content that the user is currently engaged in reading may become overwritten. There are a number of possible ways of dealing with this problem. One approach is simply to allow the user to use the *back* button in order to retrieve the previous displayed information. However, one can imagine a visitor finding such a mechanism somewhat frustrating. Another approach which which we have implemented involves the introduction of a *hold* feature. This feature enables the user to force the currently displayed infomation to remain on the screen despite any changes in context that may occur. However, in order to alert the user to the fact that the context has changed, we have incorporated a new awareness icon into the GUI in the form of a waving flag (see figure 2).

Figure 2: Alerting the user to new information awaiting presentation.

This animated icon servers to inform the user that by pressing the *update* button then new content will be displayed that reflects the changed context.

Another implication of this approach relates to the appropriateness of alerting the visitor to a change of context. In more detail, we need to consider whether the change in context is of sufficient relevance to the visitor to warrant their attention. For example, if the castle is on a visitor's tour and is closing early then it may be appropriate to make the user aware of this change. However, if the visitor has not

even expressed an interest in historic attractions via his or her personal preferences then the notification is less justified.

4. Conclusion

There is an interesting conflict that faces designers of interactive (mobile) context-aware systems. Essentialy, this conflict results from the aim to tailor or adapt the presentation of information based on the current context and the desire to build a system that adheres to the principle of least astonishment. In more detail, the designer needs to consider the importance of maintaining consistency between the user's view of their environment, the view of the environment reported by the system and the actual state of the environment.

In order to explore this conflict we are conducting a field-trial study to evaluate and subsequently compare two versions of the GUIDE system, one ostensibly based on information pull and the other based on information push. In addition, unlike the initial field trial study, this study will explore how dynamic changes in the environment can be presented to the user. It is intended that the results of this evaluation be presented at the workshop and in any final paper.

References

1. Schilit B, Adams N, Want R. Context-Aware Computing Applications. In: Proceedings of the Workshop on Mobile Computing Systems and Applications. Santa Cruz, CA, U.S. 1994.
2. Brown, P.J., Bovey, J.D., Chen, X.: Context-aware applications: from the laboratory to the market place. IEEE Personal Communications, Vol. 4, No. 5 (1997) 58-64
3. Thimbleby H. User Interface Design. ACM Press, New York, New York, 1990.
4. Long, S., Kooper, R., Abowd, G.D., Atkeson, C.G.: Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. Proc. 2nd ACM International Conference on Mobile Computing, Rye, New York, U.S., ACM Press (1996)
5. Davies, N., K. Mitchell, K. Cheverst and G. Blair. "Developing a Context Sensitive Tourist Guide", Proc. of the 1st Workshop on HCI and Mobile Devices, Glasgow, U.K., GIST Technical report G98-1, May 1998. pp 64-68
6. Cheverst K, Davies N, Mitchell K, *et al*. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In: Proceedings of CHI'00, Netherlands, April, ACM Press. 2000. pp 17-24.