'Caches in the Air': Disseminating Tourist Information in the Guide System

Nigel Davies, Keith Cheverst, Keith Mitchell and Adrian Friday

Distributed Multimedia Research Group,
Department of Computing,
Lancaster University,
Lancaster,
U.K.

telephone: +44 (0)1524 594337
e-mail: most@comp.lancs.ac.uk

Abstract

This paper describes work carried out as part of the GUIDE project at Lancaster University. The overall aim of the project is to develop a context-sensitive tourist guide for visitors to the city of Lancaster. Visitors are equipped with portable GUIDE units which in turn provide interactive services and dynamically tailored web based information reflecting the visitor's preferences and environmental context. In contrast to existing tourist systems all information in GUIDE is obtained dynamically using a city-wide wireless network infrastructure. In this paper we focus on the design of the GUIDE information model. The model presented is novel by virtue of the fact that it enables both geographic and contextual information to be captured. The paper also describes our development of an efficient broadcast mechanism which enables visitors' requests for information to be serviced quickly despite the wireless communications infrastructure employed.

1. Introduction

This paper describes work carried out as part of the GUIDE project at Lancaster University. The overall aim of the project is to develop a context-sensitive tourist guide for visitors to the city of Lancaster. These guides will take the form of hand-held units which provide information and services to their users based on individual user profiles and contextual information including the unit's physical location and other environmental factors. The overall aims are thus similar to those which motivated the work of Long et al. on the Cyberguide project [Long,96]. More generally, the work has much in common with that of Schilit [Schilit,94] and Brown [Brown,97] in the area of context-sensitive applications. However, the GUIDE project differs from these activities in a number of crucial ways:

(i) The project is developing an application which is designed to meet the real requirements of tourists as determined by a comprehensive requirements study [Cheverst,98] and which are being continually revisited through an on-going ethnographic study.

(ii) The project relies on a high-bandwidth wireless infrastructure to download information to the GUIDE end-systems as they navigate the city. This enables us to support interactive services and highly dynamic information [Cheverst,98]. In addition, this same network infrastructure provides location information to the end-systems, thus obviating the need for a separate location system such as GPS.

(iii) The project utilises an object model which combines the event-based aspects of context-sensitive models with geographic functions commonly associated with geographic information systems (GIS) and required to support our target applications.

The GUIDE system comprises a number of interconnected cells, each of which offers wireless communications coverage to a selected area of the city (see figure 1). These cells have associated *cell-servers* with local storage and processing to enable them to provide tailored information to end-systems which enter their zone of coverage. The end-systems themselves run specially designed software which provides a range of applications including information access, tourist guides and ticket booking facilities. The networked nature of the project enables us to support dynamic information such as the availability of attractions and information on weather dependent activities. Furthermore, the fact that the end-systems have network connectivity also allows us to offer interactive services such as ticket booking and accommodation reservations as well as access to information available from the web.

In this paper we describe our information model and the information propagation mechanisms that we have employed in GUIDE. Section 2 details the major findings of our requirements study into the needs of city visitors. Following this, section 3 describes our approach to modelling context-sensitive information for the city of Lancaster and in section 4 we describe in some detail a new protocol we have developed for propagating information to GUIDE end-systems. This protocol is designed to reduce client delays when accessing location-based information and exploits the broadcast characteristics of the underlying communications infrastructure. In section 5 we describe the implementation status of the project and in section 6 we discuss our future work in this area. Finally, in section 7 we present some concluding remarks.

2. Application requirements

Following a process of semi-structured interviews with members of Lancaster's Tourist Information Centre (TIC) we were able to derive a set of requirements for the GUIDE system. The three key requirements for scoping the development of the GUIDE system are described below:-

(i) Flexible Tour Guide

After studying the various demands visitors make of a tour guide it has become apparent that flexibility in this area is critical. For example, visitors to Lancaster often request city tours or trails reflecting interests as diverse as history, architecture, maritime activities, cotton production and antique dealerships. In addition, this information is required at many different levels: from academic scholar to primary school child and in a range of languages. Additional factors which affect a visitors choice of tour/trail include the geographic area to be covered, the duration of the tour, the budget of the visitor (to cover entrance fees etc.), refreshment requirements, availability of different forms of transport and any other constraints, such as wheel chair access.

(ii) Dynamic Information

The requirements study also revealed a significant requirement for the support of dynamic information. For example, Lancaster Castle is available for tours only when the court is not in session. Since this can change on an hourly basis depending on a number of arbitrary factors, e.g. a defendants' plea, such information cannot be supplied by the TIC in advance and therefore needs to be capable of dynamic change. Further examples of dynamic information

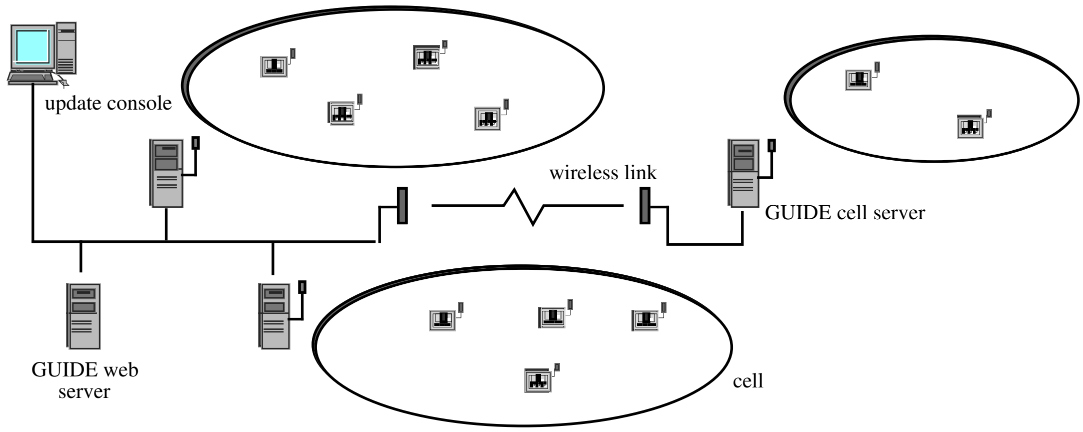

Figure 1 : The GUIDE System Architecture.

include local weather and traffic news, special events and current waiting times at local attractions.

(iii) Support for Interactive Services

The requirements study also revealed some interesting findings regarding the way in which tourists interact with the TIC. More specifically, we found that a surprising number of visitors make repeat visits to the TIC, often during the course of a single day. In most cases this is because they require additional information on activities or landmarks or they have a particular question to ask a member of the TIC. In addition, visitors would sometimes return to the TIC in order to make use of a specific service offered by the TIC, such as the booking of accommodation or travel.

The GUIDE system has been designed to address each of the above requirements. By using GUIDE end-systems, city visitors should have the ability to receive up-to-date information and access interactive services whilst touring the city of Lancaster.

3. Modelling context-sensitive information in GUIDE

3.1 Information model

The information model adopted by GUIDE was required to represent or store the following distinct types of information :-

- information which can be tailored to reflect a users context. More specifically, this is often modelled as information which can react to events such as ‘it is raining’ and ‘the user is outside’.
- geographic information, which can be either expressed in geographic terms (e.g. ‘location a is at co-ordinates x,y’) or symbolic terms (e.g. ‘location a is in the museum’) [Leonhardt,98].
- hypertext information, which can be either global (i.e. internet based) such as the world-wide-web or stored locally.
- active components, which are capable of storing state (such as a visitor’s preferences) and of performing specific actions or satisfying certain requests.

Interestingly, although each particular information type has been successfully modelled, we are unaware of any context-sensitive systems which are capable of handling the full complement of information types described above. For example, the current data models that have been designed for supporting context-sensitive information, e.g. stick-e-notes [Brown,96], are not well suited for managing geographic information. More specifically, such models require additional mechanisms if they are to be made capable of reasoning about the proximity of context-sensitive nodes and answering questions such as “what locations are near me ?”. Similarly, the data models supported by the current range of object oriented Geographic Information Systems are not well suited for representing context-sensitive information [Coyle,97]. In particular, such systems lack the necessary triggering mechanisms required for handling the events raised by changes of context.

The lack of availability of an appropriate model for satisfying the requirements of GUIDE provided the motivation for developing our own information model based on the concept of integrating an active object model with a hypertext information model. The remainder of this section describes the GUIDE information model.

The object model contains two distinct object types: objects that represent actual locations and objects representing navigation points. Location objects are the fundamental building block of our model. One example of a location object might be a specific café. This café object would have state representing its physical location within the city, opening times, the day's ‘specials’ menu and links to other nearby locations. Location objects also support methods which enable state information to be accessed and modified. For example, a selection of the methods supported by the object representing the café are:

GetImmediateNeighbours()

This method is used to return a list of object references for those objects that are immediate neighbours of the current object (in this example, the café object). Using this list of object references the GUIDE system is able to carry out route-finding and navigation tasks as well as placing context-sensitive nodes in a geographic context.

ProvideDescription()

This method is used to return a handle to information about the location object based on the visitor’s profile. For example, if the visitor happened to be a German tourist then this method would return a handle to information on the café presented in German. This provides coarse grained tailoring of the information according to user profiles. Fine grained tailoring is achieved using a separate mechanism described later in this section.

CurrentStatus()

This method enables the system to determine whether the café is currently open.

In addition, a café object would also offer a selection of specific methods allowing, for example, the dynamic modification of its menu.

The second type of object, i.e. navigation point objects, are used only to support the construction of tour guides and for route guidance purposes by representing way-points between location objects.

The object model allows geographical information to be represented by creating relationships between objects. These relationships can have attributes (weights) assigned to them to denote, for example, the distance between two objects by a variety of means of transport. Using this representation, applications (such as a tour guide object) can traverse the objects and relationships to determine, for example, neighbouring locations and optimal routes between points.

In addition to the object model described above we use a standard hypertext model to represent information in the GUIDE system. This allows information to be created and structured using well understood techniques.

The object model when combined with the use of a hypertext information repository provides an information model which meets the requirements discussed at the start of this section. The key feature of our approach is that we support a high degree of coupling between the object model and the hypertext information (as shown in figure 2).

In more detail, the object model provides a convenient mechanism for accessing hypertext pages in much the same way as a search engine enables access to web pages. More specifically, objects contain references to hypertext pages providing users and applications with multiple entry points into the hypertext information base.

Crucially, we also enable hypertext pages to reference the object model. This enables an author to construct a hypertext page which can interrogate the state of objects within the model. The page can thus be displayed differently depending on factors such as the number of times a visitor has been to a location or the current status of the attraction.

```
<P>

Welcome to <GUIDETAG INSERT POSITION> <P>

From here you can visit the neighbouring
locations of:
<GUIDETAG INSERT NEIGBOURS> <P>

<GUIDETAG INTEREST ((HISTORY > 50) AND
(ARCHITECTURE > 50))
   The following fetaures will be of
   particular interest to you ……. <P>
</GUIDETAG>
```

Figure 3 : Examples of GUIDE Tags

In order to allow hypertext pages to reference the object model we enable the authors of hypertext pages to augment their pages with tags which control the display of the information (figure 3). These tags take the form of special instructions which are able to query the GUIDE object model. For example a tag might be used to represent the time of day. If the time of day was morning then the café menu would be displayed differently to that

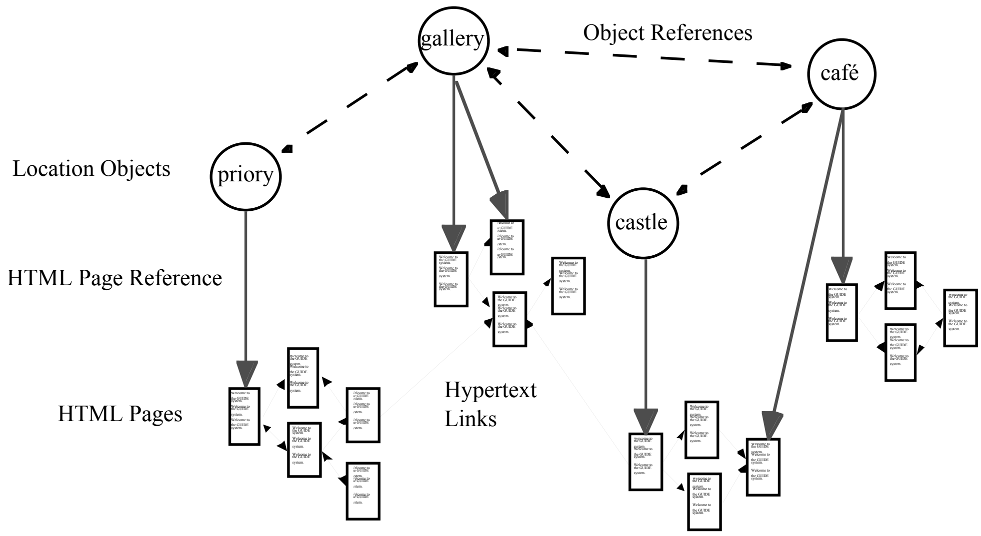

Figure 2 : The GUIDE object model.

shown if the time of day was afternoon (reflecting the fact that breakfast is only available in the morning).

The information model allows us to tailor the information we present to users without requiring the use of a specialised database to generate the information on a per-request basis. A side-effect of this is that the GUIDE system can operate using cached information without requiring continual access to an information service.

3.2. Supporting software infrastructure

The architecture used by the GUIDE system to realise the above object model comprises a number of distinct components (see figure 4).

The key component in our architecture is the GUIDE controller object. This services user requests for information and dispatches events such as position updates to the appropriate components in the system. Location and navigation objects in the object model are created and maintained by the resolver component. This maintains handles to currently active location objects and creates new location objects as required (in a similar fashion to a CORBA ORB in a distributed application).

Users access information via a HotJava browser embedded in the GUIDE application. This uses a local proxy to satisfy requests as described in detail in section 4. Pages which contain tags and which require processing are passed to the filter component prior to display. This filter component can interrogate other components during this process including the visitor's profile and arbitrary instances of location objects (via the resolver).

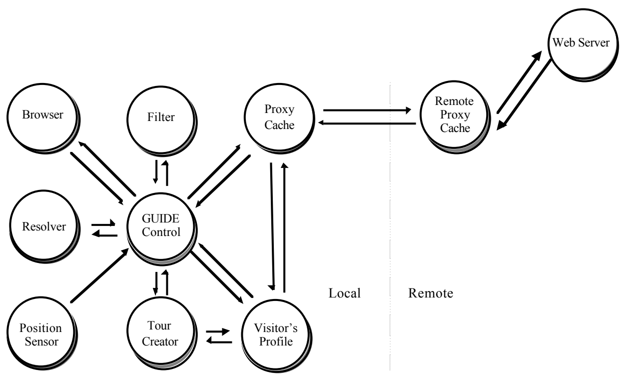

Figure 4 : The GUIDE software architecture.

Additional components are used to provide specific application functions such as the creation of tours.

The following steps illustrate the way in which objects interact in order to provide context-sensitive information to city visitors while they roam:

1) The position sensor object listens for beacons from cell servers.
2) On hearing a beacon the position sensor object notifies the control object of the visitor's new location e.g. Lancaster castle.
3) The control object requests an instance of the appropriate landmark object (e.g. the castle object) from the resolver object.
4) The resolver object returns a handle to the appropriate landmark object, creating a new instance if necessary.
5) On receiving a handle to the landmark object, the control object can invoke methods on the landmark. For example, on arrival in the castle cell, the *Arrive* method would be invoked on the Castle object. This would result in the posting of a message to the local browser object instructing it to inform the visitor of their new location and the 'visited' count in the Castle object being incremented.
6) The visitor can then use their local web browser to request information such as, 'What are the local attractions near to me ?'.
7) This request would be processed by the current location object and a handle returned to an appropriate page of information. Once this page had been obtained from the local cache or remote server it would be processed by the filter object and then displayed.

The users view of the object model can be serialised and restored at a later date to provide persistence of information.

4. Information propagation

4.1. Overview

The GUIDE system relies on the timely delivery of information to end-systems as users roam through the city. More specifically, the system must offer rapid access to a wide variety of information via the wireless network. In addition, the system must scale such that the performance of the system does not degrade unduly as the number of users in each cell increases. Since our wireless cells may be relatively large (see section 5 for details) this implies that a fully functioning GUIDE system would be aiming to support in the order of 100 users in the busiest cells.

Since most of the information in GUIDE is stored as web pages this, at first sight, appears to be the now familiar problem of reducing web access times. However, since GUIDE is a self-contained, single application system, the reduction in access times is the sole objective. This may be contrasted with conventional web systems where the reduction in access times must be traded off against network usage. As a result, we are able to utilise network bandwidth in order to reduce access times by employing a broadcast based approach to information dissemination [Franklin,96], [Zdonik,94].

The overall GUIDE system may be viewed as a central web server that mobile clients access via the WaveLAN network. In order to improve the performance of the system we place caches in each cell and user requests are, where possible, satisfied by this local cache.

Additionally, we periodically broadcast a subset of the contents of each cell's cache to users within the cell. In this way, users who enter a cell receive (and cache on their mobile unit) frequently requested pages for the cell. This approach has a number of benefits:

1. *Improved response times*

Many of the pages which users request will already have been cached on their mobile unit and hence response times will be extremely rapid since only local access will be required.

2. *Improved scalability*

Since clients receive many of the pages they require without transmission the system scales well as the number of end-systems increases. The exact extent to which the system scales depends on the pattern of user requests and this is discussed further below.

3. *Power saving*

The WaveLAN cards we employ in GUIDE utilise considerably less power when receiving than when transmitting. By reducing the need for clients to transmit requests there should be a corresponding decrease in power consumption.

The precise benefits of this approach depend on a number of factors. Most crucially, in a situation where there is a large degree of uniformity of page requests (i.e. most users request the same subset of available pages) the system scales extremely well to support large numbers of users. As the uniformity of requests diminishes so the gains over a conventional request-reply system are reduced. In the worst case (no commonality) the system will perform worse than a conventional system since time will be spent re-broadcasting pages which are not required.

While ensuring the conformity of user requests would be a significant problem in a general purpose system, we believe that in GUIDE a user's access patterns will be largely dictated by their physical location. As a consequence, we expect each cell's cache to gradually build up a broadcast schedule containing information relating to the physical location of the cell.

Additional factors which often affect the performance of broadcast based systems are the degree to which parameters such as the number (length) of pages broadcast, the frequency with which pages are broadcast and the strategy for replacing pages in the broadcast schedule can be tuned to match application requirements [Imielinski,94].

It is our opinion that these factors are less critical for GUIDE than for most other broadcast based systems. In particular, our key objective is to reduce the time the user perceives they are waiting for pages, i.e. the number of pages which have not already been cached when the user attempts to access them. If we consider that each end-system has an in-memory cache of, for example, 6 MB then this is ample to cache almost all of the information relating to a given cell (the exception being video clips which we do not currently support). Over a WaveLAN network it would take approximately 30 seconds to completely fill the cache: substantially less time than it would take most users to read the first page of information relating to a given geographic area. Furthermore, we can delay presenting information to a user about a new cell until sufficient pages are cached since the user has no means of telling that they have entered a new cell until they are informed of the fact by the application. Hence, assuming a user does not deviate substantially from the normal access patterns they will experience almost no delay in accessing information. The relatively high network bandwidth, large caches and predictable access patterns found in GUIDE conspire to make fine tuning of the classic broadcast strategy parameters unnecessary.

4.2. Engineering issues

We have engineered our broadcast based caching system using proxies which run on the mobile end-system and the cell servers. The server-side proxies build up lists of pages to broadcast and periodically transmit these pages within their cell. An index of the pages to be broadcast is appended to the start and end of each transmission so that mobile units are able to determine the current contents of the broadcast cycle. All of the information is multicast to a well known IP multicast address.

Client-side proxies listen for broadcasts and fill up their caches with the contents of the latest broadcast. If a client machine enters a cell midway through a broadcast it will only partially fill its cache but will be able to determine from the trailing index those pages which it has missed but which will be retransmitted next cycle.

User requests for pages are all routed via the client-side proxies. If the request cannot be satisfied from the current cache contents and the client does not believe it is scheduled for transmission soon (determined from the index) then it can issue a request for the appropriate page. This is received by the server proxy which will fetch the page if it is not already cached locally and then schedule the page for transmission in the next cycle.

We are not running mobile IP or WaveLAN roaming and hence clients which require only those pages contained in the current broadcast cycle need never transmit to the servers.

5. Implementation status

5.1. End-system selection

We have selected the Fujitsu TeamPad 7600 as the GUIDE end-system. This is a compact unit measuring 8"x9"x1.5", is based on a 486 100 MHz processor and has been ruggedised to withstand drops from approximately four feet onto concrete. The relatively poor performance of the processor is of little concern to us since we are not anticipating running CPU intensive tasks. Furthermore, the slow speed of the processor enables the unit to function for up to four hours on a single 3"x2"x1.5" battery. Further details on the TeamPad can be found at [Fujitsu,98].

5.2. Application development

The GUIDE system comprises the following software components: a standard Apache HTTP server running under Linux, Java cache, beaconing and broadcasting applications running on the cell-servers and a GUIDE client application which supports the information model described in section 3.

Figure 5 : The Guide Prototype.

The GUIDE client software is based on current WWW technology and the interface has been constructed using the HotJava HTML Component.

Figure 5 illustrates the main features of the current prototype. The text message box in the bottom left displays the user's current location and is updated as a result of the position sensor receiving a new beacon. The main window is used to display information and, in this example, shows a web page welcoming the user to the current location. The page has been tailored to reflect the fact that the user has not previously visited the location and is physically at the site (rather than taking a virtual tour). The buttons at the top of the screen offer access to local information and general page navigation facilities and the buttons at the bottom of the screen allow for the invocation of interactive services such as making a cinema ticket reservation or sending a message to another user or groups of users.

5.3. Cell deployment

We are in the process of installing cells in the city of Lancaster. Each cell consists of an HP Vectra 6/200XU acting as a cell sever, a WaveLAN ISA card for broadcasting information to the cell and a link back to the University network. This link is via either a leased line or, where this is not available, the EDNET wireless network [Forde,97] which has been deployed by Lancaster University to provide internet access to local schools.

Our experience is that the WaveLAN cards provide coverage with a radius of approximately 200m in free space. However, the signal has very poor propagation characteristics through buildings and hence in a city the actual cells tend to be bounded by the city architecture. Within the context of GUIDE this is a positive feature since it allows us to create smaller, non-overlapping cells and hence provide more accurate position information and avoid the need to support WaveLAN roaming. However, we have not found it possible to create very accurate cell boundaries due to the reflection of signals, interference and changes in propagation patterns resulting from fluctuations in the density and placement of objects within the cell. For example, where we have placed cells in city squares the degree of propagation down the square's access roads varies with the prevailing conditions.

We have currently deployed five cells and hope to deploy a further five in the near future in order to provide coverage of many of the main tourist attractions in Lancaster.

6. Future work

6.1. User interface design

GUIDE end-systems are required to react to changes in their environment. In particular, as users move around the city of Lancaster they require information which is relevant to their physical location. Furthermore, this information may replace existing information with which the user has previously been presented. The implication of this for the design of the user-interface is significant since it raises the problem of integrating changes in physical location with changes in information within the system

[Davies,98]. For example, if the GUIDE system provides a button labelled local attractions one might intuitively expect this to provide information relative to the area in which the user is located. However, users accustomed to web based systems are likely to be confused by a system in which returning to previously visited pages does not provide the same information. Furthermore, since a user might wish to check out the attractions local to their destination rather than their current location a means of simulating changes in their physical location must be included and supported by appropriate navigation tools. The development of suitable user-interface metaphors for context-sensitive information applications will form a major focus of our future work.

6.2. Evaluation

The GUIDE project is scheduled to conduct field trials of the completed system early next year (1999). However, the evaluation of the system is problematic both from performance and user acceptance perspectives.

With respect to performance, the key issue is that the main system measurement, i.e. response time, is heavily dependent on user behaviour. More specifically, the time users have to wait to access information in the GUIDE system will depend both on the number of users of the system and the diversity of requests issued by these users. We are unlikely to be able to deploy sufficient end-systems during the field trial to be able to test the scalability of the system and hence will be looking to develop simulation models of the system. However, the development of these models is heavily dependent on our obtaining suitable data concerning typical usage patterns of the system and it is these usage patterns that we hope to obtain from our early field trials. Once we have an accurate picture of the usage patterns of the GUIDE system we will investigate in more detail alternative information dissemination models such as those proposed by Wong in [Wong,98].

Evaluating GUIDE from a user acceptance perspective is also likely to prove difficult. In particular, we would like to observe people using and interacting with the system during the course of their visit to Lancaster without, of course, unduly influencing their behaviour. We do not propose to carry out such an evaluation ourselves but, instead, hope to utilise the skills of the HCI community in this area and we are in on-going discussions with a number of groups interested in conducting such an evaluation.

7. Conclusion

The GUIDE project is developing context-sensitive tourist guides for visitors to the city of Lancaster. The novel feature of the project is that information is downloaded to visitors via a high bandwidth wireless network as they navigate the city. This enables us to support dynamic information and interactive services, both of which we have identified as important requirements of such systems.

In this paper we have focused on the modelling and propagation of context-sensitive information within the GUIDE system. More specifically, we have presented an information model which combines aspects of current context-sensitive information models with features traditionally associated with geographic information systems. This model enables us to create objects which respond to events, and hence can tailor their behaviour to reflect their context, and to reason about the geographic relationships between these objects. The object model provides access points into a collection of web pages. The presentation of these pages can be dynamically tailored to reflect a user's current context through the use of tags which enable page authors to interrogate the object model at run-time. This approach provides a powerful set of tools which enables information to be easily tailored to support context-sensitive operation.

The propagation of information within the GUIDE system is handled by a custom broadcast protocol which ensures rapid response to user requests despite the use of a wireless network for information access. This is combined with caching on the mobile units in order to allow them to continue to function during periods of disconnection.

We have completed the design and implementation of the main GUIDE components and are in the process of deploying cells within the city of Lancaster. We hope to conduct field trials with members of the public in early 1999 and to use the results of these field trials to drive further refinements to the GUIDE architecture.

Acknowledgements

This work was carried out as part of the EPSRC funded GUIDE project (GR/L05280) in co-operation with Lancaster City Council and has received support from Lucent Technologies.

8. References

[Brown,96] Brown, P.J. "The stick-e document: a framework for creating context-aware applications." Proc. EP'96, Palo Alto, U.S., Pages 259-272. 1996.

[Brown,97] Brown, P.J., J.D. Bovey, and X. Chen. "Context-aware applications: from the laboratory to the market place." IEEE Personel Communications, Vol. 4 No. 5, Pages 58-64. 1997.

[Cheverst,98] Cheverst, K., N. Davies, K. Mitchell and G.S. Blair. "The Design of an Object Model for a Context-Sensitive Tourist Guide." Proc. Interactive Applications of Mobile Computing (IMC'98), Rostock, Germany, Pages 25-29. 1998.

[Coyle,97] Coyle, M., S. Shekhar, D. Liu and Sarkar S. "Experiences with Object Data Models in Geographic Information Systems". *Communications of the ACM* , Vol. 40, No. 4, Pages 103-111. April 1997. (Tech. Report TR95-10, Univ. of Minnesota, Computer Sc. Dept.).

[Davies,98] Davies, N., K. Mitchell, K. Cheverst, and G.S. Blair. "Developing a Context-Sensitive Tourist Guide" Proc. 1st Workshop on Human-Computer Interaction for Mobile Devices, Glasgow, U.K., Pages 64-68. 1998.

[Forde,97] Forde, B., "Lancaster's EDNET Linking Lancaster's Schools to the Internet", Panel presentation at MobiCom97, Budapest, Hungary, September 26-30, 1997. Source for slides available at: http://148.88.16.66/techinfo/MobiCom97/index.htm.

[Franklin,96] Franklin, M., and S. Zdonik. "Dissemination-based Information Systems." IEEE Data Engineering Bulletin, Vol. 19 No. 3, 1996.

[Fujitsu,98] Fujitsu TeamPad Technical Page. http://www.fjicl.com/TeamPad/teampad76.htm.

[Imielinski,94] Imielinski, T., S. Vishwanathan, and B.R. Badrinath. "Energy Efficiency Indexing on Air." Proc. International Conference on SIGMOD 94, Miniapolis, U.S., Pages 25-36. 1994.

[Leonhardt,98] Leonhardt, U. "Supporting Location-Awareness in Open Distributed Systems" PhD Thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London. 1998.

[Long,96] Long, S., R. Kooper, G.D. Abowd, and C.G. Atkeson. "Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study." Proc. 2nd ACM International Conference on Mobile Computing (MOBICOM), Rye, New York, U.S., Pages 97-108. 1996.

[Schilit,94] Schilit, B., N. Adams, and R. Want. "Context-Aware Computing Applications." Proc. Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, U.S. Pages 85-91. 1994.

[Wong,98] Wong, T., T. Henderson, S. Raman, A. Costello, and Randy Katz. "Policy-Based Tunable Reliable Multicast for Periodic Information Dissemination." Submitted to WOSBIS'98

[Zdonik,94] Zdonik, S., M. Franklin, R. Alonso, and S. Acharya. "Are Disks in the Air Just Pie in the Sky?" Proc. Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, U.S., Pages 12-20. 1994.