PERGAMON
Computers & Graphics 23 (1999) 883–891
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

IMC '98

Design of an object model for a context sensitive tourist GUIDE

Keith Cheverst*, Keith Mitchell, Nigel Davies

Distributed Multimedia Research Group, Department of Computing, Lancaster University, Lancaster, LA1 4YR, 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. The paper presents an overview of the GUIDE system, with particular focus on our design of an appropriate object model, capable of handling the high degree of flexibility required by GUIDE. In addition, the paper includes the results of our requirements analysis, the web-based application that has been developed to meet these requirements and our approach towards communicating information to portable GUIDE units in a timely and scaleable manner. © 1999 Elsevier Science Ltd. All rights reserved.

1. Introduction

The prospect of providing city visitors with dynamic and context-sensitive multimedia information whilst on the move is both novel and exciting and raises many interesting research issues. The GUIDE project [1–3] is currently examining these issues by developing systems and application-level support for hand-portable multimedia GUIDE units with wireless network connectivity, enabling context-sensitive information to be presented to city visitors whilst they roam.

The support for context-sensitive information by GUIDE has been strongly influenced by the work of Schilit [4] and Brown [5]. In essence, the notion of context is required in order to provide GUIDE units with the *intelligence* to adapt their behaviour in order to meet both the preferences of the visitor and the state of the visitor's current environment.

At an early design stage, the fundamental decision was made to base the GUIDE system around a distributed cellular architecture, comprising a number of strategically located base stations. This architecture enables information to be broadcast to a portable GUIDE unit, once it enters an area of cell coverage. For reasons of scalability and because of the limited amount of bandwidth available in any one cell, it was decided that some form of broadcast-based approach to data dissemination should be used for transferring information to the portables units. Each base station being responsible for broadcasting information that is largely specific to its own geographic location. However, because of the need to enable visitors to request specific information, including information held on the internet, each broadcast schedule shall also have some number of user request slots. Fig. 1 illustrates this basic GUIDE architecture and shows how the GUIDE system utilises web-based technologies in order to disseminate information to mobile GUIDE units. In effect, the overall GUIDE system may be viewed as a central web server that mobile clients access via the WaveLAN wireless communications network.

An alternative to the network-based approach, would have been to produce some form of stand-alone system in

* Corresponding author. Tel.: + 44 (0)1524 594539.
E-mail addresses: kc@comp.lancs.ac.uk (K. Cheverst), mitchelk@comp.lancs.ac.uk (K. Mitchell), nigel@comp.lancs.ac.uk (N. Davies)

0097-8493/99/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved.
PII: S0097-8493(99)00119-3

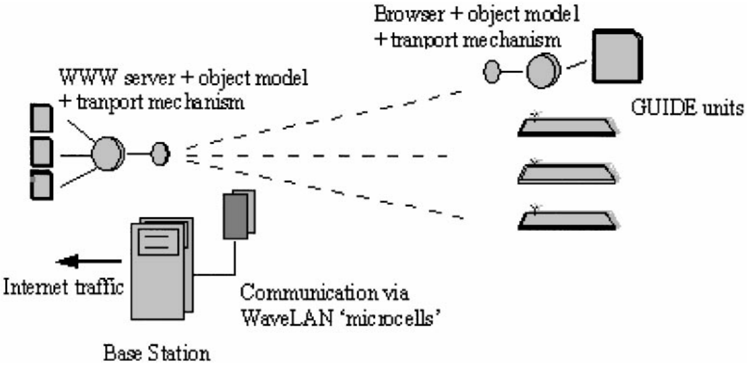


Fig. 1. The GUIDE Architecture.

which the entire working set of information is stored on the end system. Such an approach was adopted by the Cyberguide project [6]. This approach is, however, unsuitable for supporting the flexibility required by GUIDE. For example, the stand-alone approach cannot support dynamic changes to information. So, although such a system would be capable of presenting information based on a visitor's changing geographic coordinates, the information displayed could have become out of date since the units working set was loaded. Another problem with this approach, concerns the size of the working set which would need to be stored. A truly useful visitor's guide should, for example, contain information stored in different languages, for different reading ages and assuming different levels of background knowledge for each location. The storage requirements for storing all the different possible permutations would be considerable and beyond the capacity of the current generation of portable machines. One further reason for rejecting this approach was based on the fact that, in the medium term, portable web-client-based machines will become available which, compared to stand-alone PCs, will consume less power and be cheaper.

In this paper we present the results of our requirements analysis and development work on the GUIDE system. In particular, we focus on our design of an appropriate object model, capable of handling the high degree of flexibility required by GUIDE.

2. Requirements

Following a series of semi-structured interviews with members of Lancaster's Tourist Information Centre (TIC), we obtained four key requirements for an intelligent visitor guide, namely: (i) the system needs to be flexible, (ii) the information presented by the system should be context-sensitive, (iii) the system should support dynamic information and (iv) the system should offer access to interactive services.

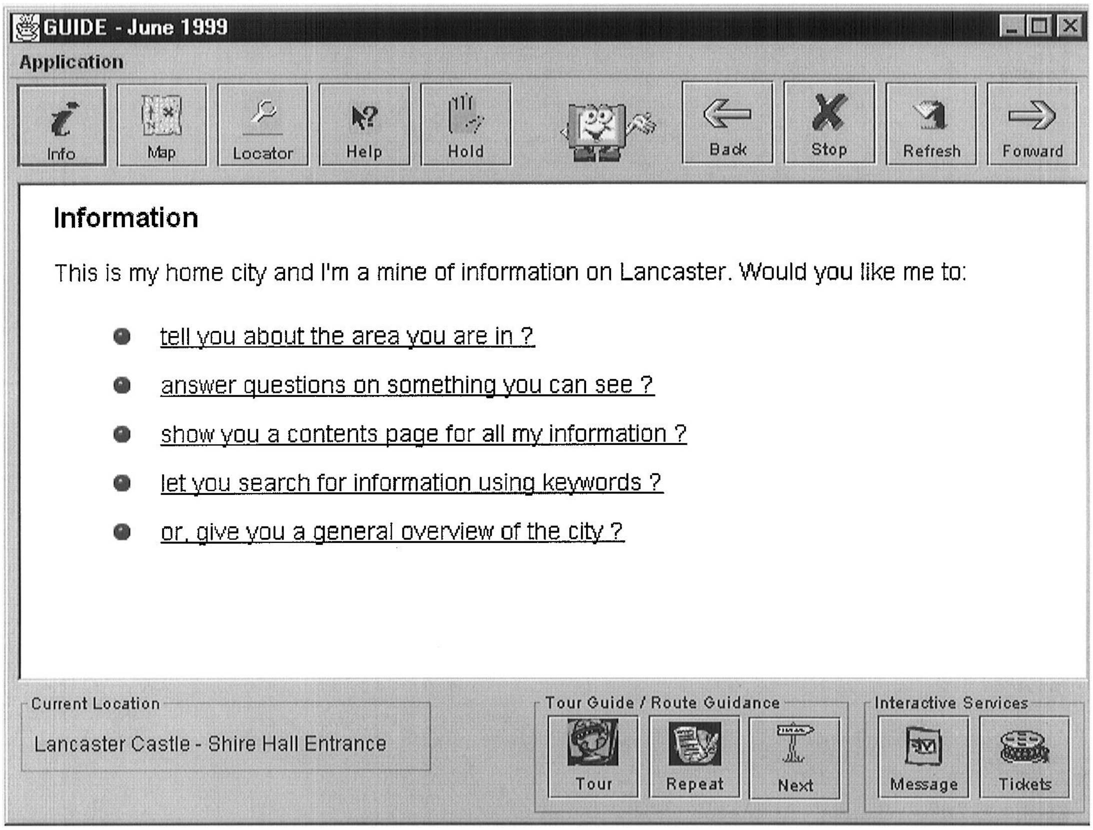
This set of requirements and examples of the way in which the GUIDE application meets each requirement is described below:

2.1. Flexibility

One key requirement for the GUIDE system is that visitors should be able to use the system as much or as little as they desire. For this reason, the GUIDE application presents visitors with a variety of choices for using the system, e.g. enabling the visitor to search the system for specific information using a keyword search facility or enabling the visitor to prompt the system for information regarding the current location. Fig. 2 illustrates the user interface presented to the visitor in order to enable them to select from various options (i.e. hypertext links) on the currently displayed page or to select an option by touching one of the buttons on the customised browser.

By selecting the tour button shown in the bottom centre of the screen, the visitor can request the system to create a tailored tour of the city by touching. A 'Guide Tour Wizard' is provided to enable a visitor to quickly and easily select those attractions she particularly wishes to visit. The system is capable of directing visitors from one location to the next, providing relevant information throughout the tour. For example, Fig. 3 demonstrates the way in which the GUIDE system presents the city visitor with both detailed textual descriptions and pictures in order to assist the visitor in reaching the next part of the tour.

In addition to receiving directions by description, we have found that some users prefer to be presented with an annotated map showing the direction in which they should travel to reach the next location. The GUIDE



Information

This is my home city and I'm a mine of information on Lancaster. Would you like me to:

- tell you about the area you are in ?
- answer questions on something you can see ?
- show you a contents page for all my information ?
- let you search for information using keywords ?
- or, give you a general overview of the city ?

Current Location: Lancaster Castle - Shire Hall Entrance

Fig. 2. The GUIDE information page.

application provides such a facility and an example of the kind of map presented to the visitor by the GUIDE system is shown in Fig. 4.

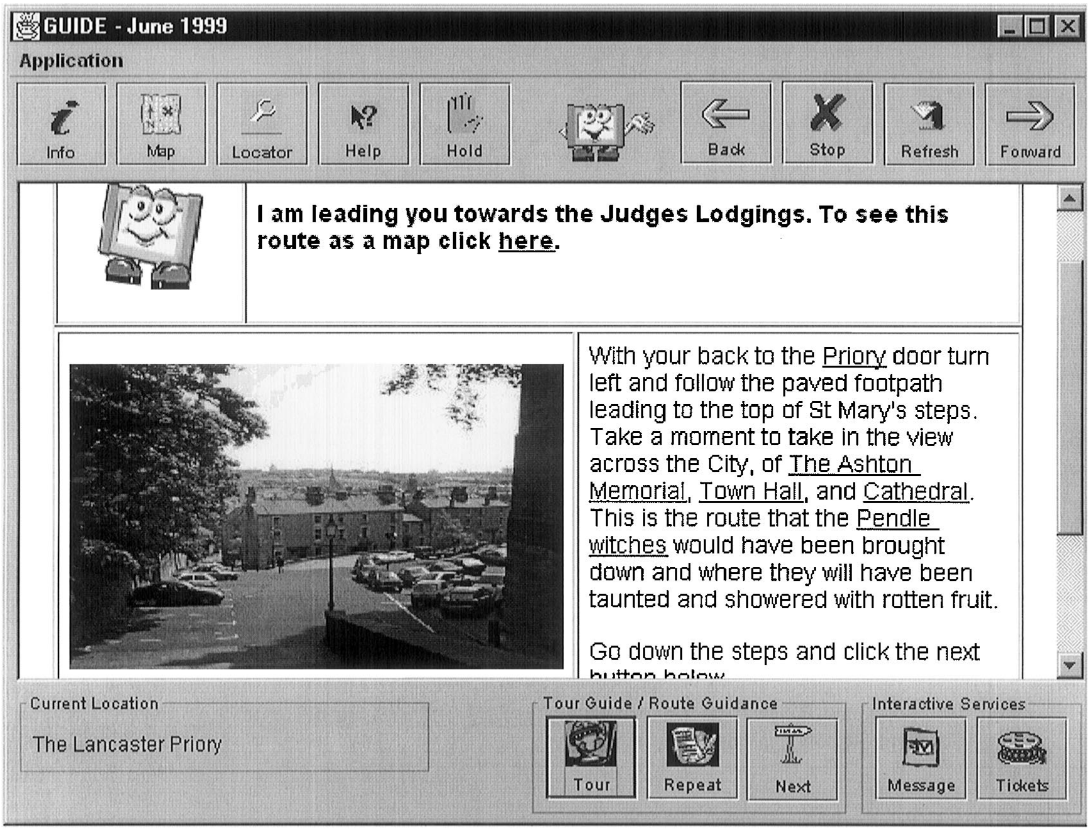
If a visitor does not wish to be taken along a tour but would rather be left to roam the city, but still have the ability to ask the system for specific pieces of information along the way, then the visitor can simply opt not to create a tour and can interact with the system using the interface shown in Fig. 2.

2.2. Context-sensitive information

A further requirement for GUIDE was that the information presented to visitors should be context sensitive. There are two classes of context that should be used, namely personal and environmental. Perhaps, the most significant piece of personal context is the visitor's interests. Such interests can be as diverse as history, architecture, maritime activities, cotton production and antique dealerships. Additional personal context that should be used includes: the visitor's current location, the amount of time they wish to spend on their visit, their budget (to cover entrance fees, etc.) and any refreshment preferences they might have. Examples of environmental context to be used as context include: the time of day, the weather, the season and the state of the city's transport system, e.g. the location of traffic congestion or the closure of walkways. If a visitor has requested a tour of the city, then the GUIDE system should use both personal and environmental context to create a suitably tailored tour. Context should also be used when presenting information to the city visitor. For example, information should be presented in a way that is suitable given the age and technical background of the visitor and there preferred reading language.

2.3. Support for dynamic information

During our study we found there to be a significant requirement for the support of dynamic information. Such information should be made available to visitors whenever their context deems this to be appropriate. For example, consider the hypothetical scenario in which a visitor touring the city has expressed a particular



I am leading you towards the Judges Lodgings. To see this route as a map click here.

With your back to the Priory door turn left and follow the paved footpath leading to the top of St Mary's steps. Take a moment to take in the view across the City, of The Ashton Memorial, Town Hall, and Cathedral. This is the route that the Pendle witches would have been brought down and where they will have been taunted and showered with rotten fruit.

Go down the steps and click the next button below

Current Location: The Lancaster Priory

Fig. 3. The use of text and pictures to guide the visitor.

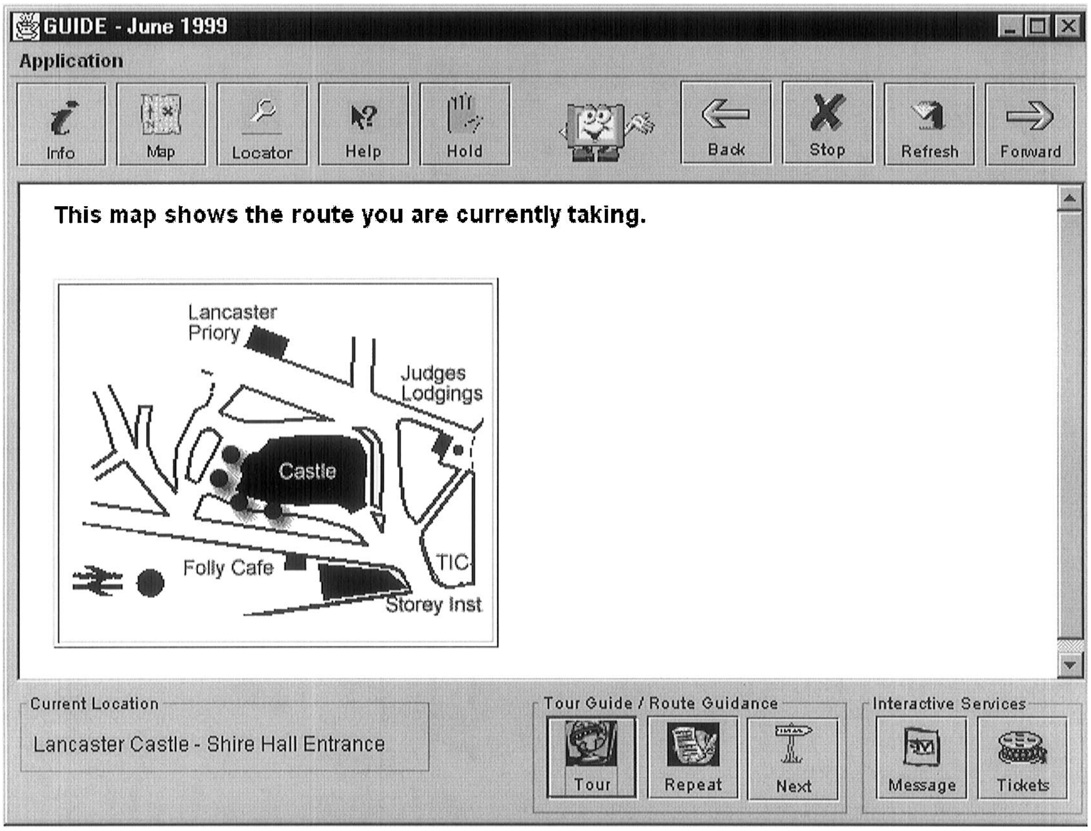
interest in Lancaster castle. When the visitor started their tour, the castle was closed to the public because the court room, situated within the castle, was in session. The GUIDE system would, therefore, not have included a visit to the castle in its recommended tour. However, if the court session finished early then the GUIDE system should use this dynamic information to notify the visitor that the castle is now open to the public and optionally modify their tour schedule to include a visit to the castle. Further examples of dynamic information include changes in the weather and average waiting times at local attractions.

2.4. Support for interactive services

Studying tourist activities in Lancaster revealed 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, they have specific questions which require interaction with a member of the TIC, or they wish to make use of a service offered by the TIC, most commonly the booking of accommodation. In addition to supporting interactive services based around the TIC, the GUIDE system also provides support for other interactive services, such as the remote booking of cinema tickets and accommodation as shown in Fig. 5.

3. Object model

A number of interesting issues arose during our design of an object model to satisfy the requirements described above. One issue being how to manage the great variety of information which the GUIDE system should be capable of providing to city visitors. Our initial approach to this problem, was to create a large set of static HTML pages, some of which would be stored locally and others would be available via remote web servers. The problem with this approach, was that it involved trying to create a set of pages to match each possible permutation of personal and environmental context for each and every location in the city. So, for example, a different HTML page would be served for the Castle location if the visitor's main interest was refreshments to that served if the visitor's main interest was medieval architecture. Clearly,



This map shows the route you are currently taking.

Lancaster Castle - Shire Hall Entrance

Fig. 4. The use of annotated maps to guide the visitor.

the two pages would contain a large degree of commonality, e.g. pictures of the castle and a summary of the castle.

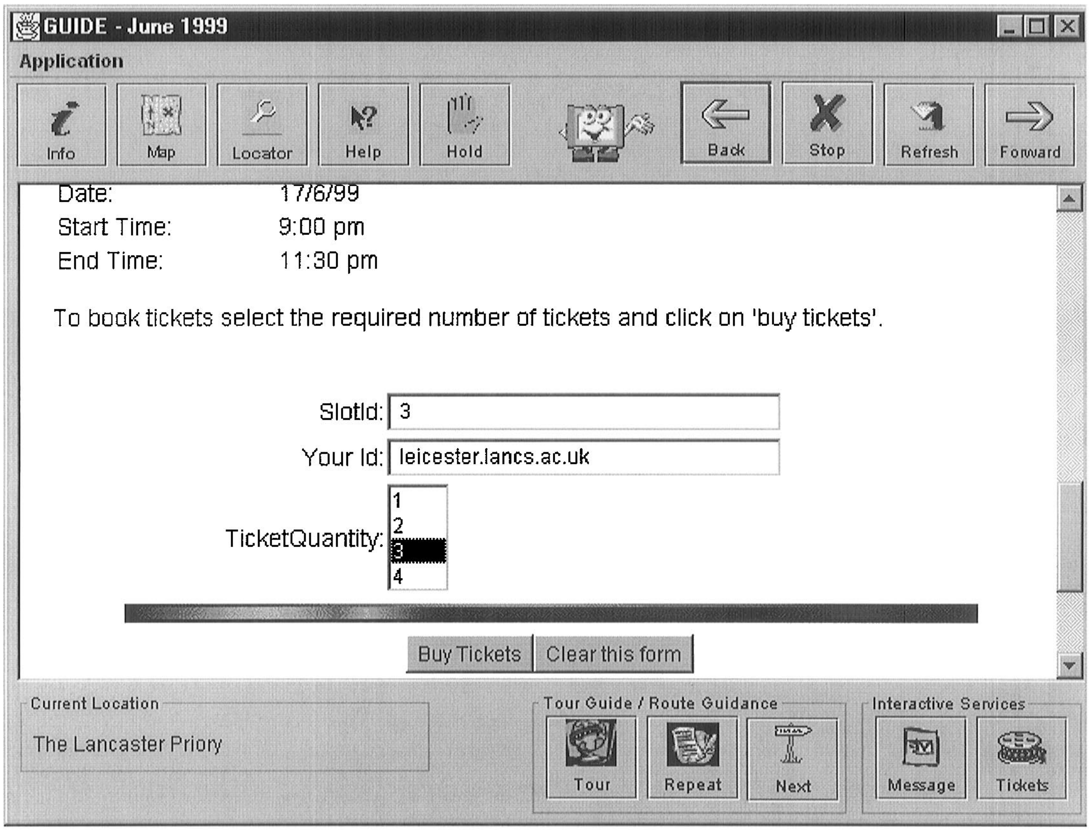
In an attempt to make use of this commonality, we adopted an approach which essentially involves the fusing together of HTML-based packets of information, in order to dynamically compose the required HTML pages. If we consider the castle-based example, then the packets of information would include various pictures of the castle and the nearby café and various pieces of text, including a summary of the castle, a description of the castle's architecture and the menu offered by the nearby café.

The actual interaction between the components of the object model are illustrated below in Fig. 6. It is worth noting that all components of the object model are Java based.

A visitor interacts with their GUIDE unit through their own local web browser. All HTTP requests are processed by the local web server object, which may in turn need to interact with other objects in order to service the request. Should the visitor request other services, such as the creation of a tour, then the local web browser invokes the services provided by the GUIDE control object. This object is responsible for controlling and communicating with the various other system components, namely, the local position sensor object, the tour creator object, the resolver object and the visitors profile object. The control object is also responsible for storing the visitor's current location.

The function of the local position sensor object is to notify the control object, via its *receiveNewPosition* method, whenever the city visitor enters a new or previously visited location. A resolver object is then responsible for creating new instances of landmark objects based on the visitor's current preferences, which are stored in the visitor's profile object. This object supports a number of methods for returning the visitor's details, e.g. the *returnPreferredLanguage* method, and these methods are invoked by either: the control object, the tour creator object or the local web server object.

The local web server object, interacts with the control and visitor's profile objects in order to obtain the context for dynamically creating a tailored HTML page formed from a collection of information packets. The local web server also interacts with a local proxy object in case some or all of the required information packets have been



Date: 17/6/99
Start Time: 9:00 pm
End Time: 11:30 pm

To book tickets select the required number of tickets and click on 'buy tickets'.

SlotId: 3
Your Id: leicester.lancs.ac.uk
TicketQuantity: 1 2 3 4

The Lancaster Priory

Fig. 5. Booking cinema tickets via GUIDE's Interactive Services.

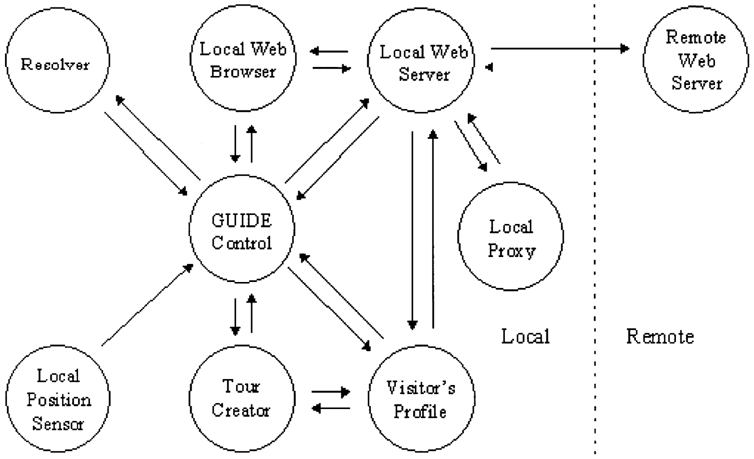


Fig. 6. The GUIDE object model.

cached. If some of the packets are not available in the local cache then a request is made for the relevant packets to a remote web server using a free slot in the broadcast schedule.

The design of this object model satisfies the requirements for flexibility, context sensitivity, support for dynamic information and support for interactive services. The following steps illustrate the way in which objects interact in order to provide context sensitive information to city visitors whilst they roam:

(1) The position sensor object listens for beacons from remote base stations.
(2) On hearing a beacon the position sensor object notifies the control object that a beacon has been heard. The control object checks the user's current position to the beacon heard.
(3) If the beacon heard suggests the user has entered a new location, e.g. the castle, then the control object instantiates the appropriate landmark object (e.g. the castle object). This is known as the active object. A landmark object stores state information (such as geographic position and whether the location has been visited previously) and supports methods to modify this state information and post messages to the control object. The control object then queries the active object for it's neighbours and instantiates those objects also.
(4) Once the control object has a handle to the active and neighbouring objects, the control object can invoke methods on those objects. These methods can access or modify the state of an object. For example, if a user is searching for a café in a particular area, the objects can be queried by the GUIDE search engine to look for references to particular keywords or interests. The visitor can then use their local web browser to request information such as, 'What are the local attractions near to me?'.
(5) This request would be served by the local web server object which would obtain the required information (based on the visitor's context) from the local proxy object, or from the remote server object, if some or all of the information was not available locally.

The design of the object model is such that the integration of new features to the GUIDE system should be relatively straightforward and not require the rewriting of large sections of application code.

A standard hypertext model is used to represent information in the GUIDE system and this enables information to be structured using standard and well understood techniques. A strong level of coupling exists between the hypertext model and the object model and this relationship is illustrated in Fig. 7.

Interaction between the object model and the hypertext model is achieved by providing location objects with references to hypertext pages. This enables applications to access the hypertext information base via multiple entry points. In addition, hypertext pages can reference and interrogate the state of objects within the object model. This functionality enables information, i.e. hypertext pages, to be displayed differently based on certain aspects of a location object's status, e.g. the number of times that a visitor has visited a location. This is achieved by using special tags to query the object model. Such tags are embedded in the HTML code of hypertext pages and provide some control over the display of information. For example, a tag is used to represent the number of times that a visitor has visited the castle. Thus, when displaying the hypertext page associated with the castle, the value of this tag can be checked and if it is greater than zero then the page can include a message stating that the visitor has visited the location previously.

4. Communications approach

The communications support for GUIDE was developed to address the following requirements:

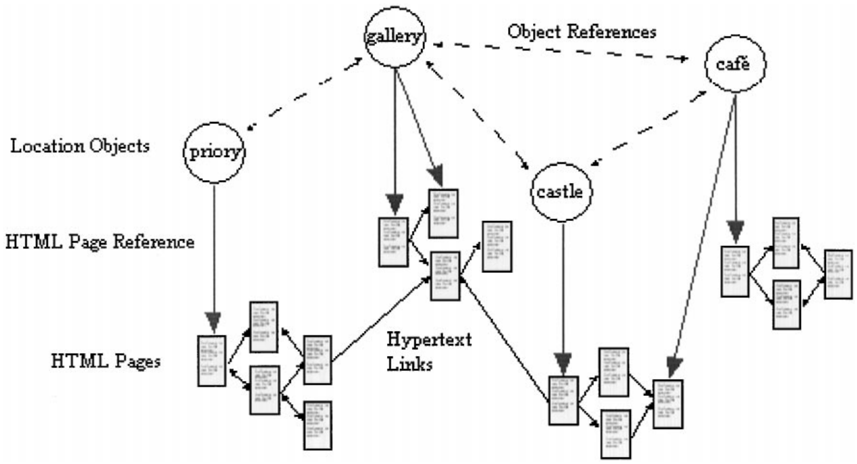


Fig. 7. The GUIDE object model.

Scalability: The system must be capable of supporting a potentially large user community requiring access to data simultaneously, therefore the system must adequately scale.

Rapid response times: Following on from the previous requirement, the system should be able to respond to a visitor's request for information in a rapid manner even when the number of visitors using the system is high.

Flexibility of services: The system should support both data broadcast and interactive services. Thus, it must provide a high bandwidth down link channel for the broadcast of data and also include spare time-slots within the broadcast schedule to enable clients to make explicit requests for information or services.

Support for Disconnected operation: There will be areas of disconnection across the city of Lancaster which should not overly disrupt the services provided by GUIDE as visitors roam around the city.

Minimal usage of power: The fact that battery life is still a major limitation with portable end-systems necessitates that the power drain associated with supporting communications be kept to a minimum.

To meet these requirements we have chosen to replace TCP as the means of communication between the mobile units (clients) and base stations (servers) in favour of a broadcast-based protocol. This protocol builds on the previous work on broadcast disks as a means of disseminating data [7,8]. In more detail, 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.

All user requests for pages are routed via client-side proxies. In the event of a request not being satisfied from the current cache then the client will determine whether the requested page is scheduled for transmission soon. If it is then it will await the transmission, otherwise it will issue an explicit request for the page. This explicit request will be 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.

This broadcast-based approach is well suited for meeting the requirements described above. More specifically, the approach fulfills the requirement for scalability by enabling the system to support a large number of clients within each network cell by utilising the 2 Mbit/s of available bandwidth per cell. In addition, the system achieves extremely rapid response times in the majority of cases because, in general, the pages which visitors request will already have been cached on their mobile unit and therefore only local access will be required. The adopted approach is sufficiently flexible to support both the broadcast of data to mobile units and (via the use of free broadcast cycles) and interactive services. The fact that mobile units cache information locally enables units to provide some useful functionally even when operating outside of communications coverage. The requirement for minimising the use of power for communication purposes is also fulfilled by the broadcast based approach because the WaveLAN cards utilised by GUIDE use considerably less power when receiving than when transmitting. The use of a broadcast approach means that, in general use, clients rarely have to make transmissions and as a result the power drain on the portable end-systems due to communications should be kept to a minimum.

It is, however, important to note that the benefits of utilising a broadcast-based approach actually depend on a number of factors. For example, 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) then the system scales extremely well to support large numbers of visitors per cell. However, as the uniformity of requests diminishes so the gains over a conventional request-reply system are reduced. Indeed, in the worst case, i.e. where there is no commonality of page requests, the system will perform worse than a conventional system since time will be spent re-broadcasting pages which are not required. For the GUIDE system, we predict that a visitor's access patterns will be strongly influenced by their physical location. Consequently, we expect each cell's cache to gradually build up a broadcast schedule that contains information relating to the physical location of the cell.

5. Current status

5.1. End systems

We considered a wide range of end-systems for use in GUIDE, including pen-based tablet PCs and PDAs, and finally selected the transflective version of the Fujitsu TeamPad 7600. This compact, ruggedised, unit measuring 8″ × 9″ × 1.5″, is based on a 486 100 MHz processor. Its transflective, greyscale, screen enables the display to be readable even in direct sunlight, which is currently not possible with colour screens. Further details on the TeamPad can be found in [9].

5.2. Communications infrastructure

We have currently deployed seven communications cells in the city of Lancaster and hope to deploy further three in the near future in order to provide coverage of many of the main tourist attractions in 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 [10] which has been deployed by Lancaster University to provide internet access to local schools.

The WaveLAN cards provide coverage with a radius of approximately 200 m in free space. However, the signal has very poor propagation 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. Unfortunately, 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.

6. Concluding remarks

In this paper we have described our on-going development of a context sensitive intelligent guide for city visitors. The requirements for such a guide have been presented and we have outlined our design for an object model capable of providing the high level of flexibility required by the GUIDE system. In particular, the model supports the provision of dynamic and context sensitive information by adopting an approach based on the composition of dynamically tailored web pages from a collection of both locally, and remotely, stored information packets.

The GUIDE project is currently deploying its infrastructure throughout the city of Lancaster and initial feedback following the evaluation of the system by other members of our department has been very positive. The system is rapidly approaching the stage at which it can be demonstrated and evaluated by members of the public. This evaluation is likely to prove difficult, not least because of the need to observe city visitors interacting with the system without affecting their behaviour. In addition, we feel it important that a large cross section of people are used to evaluate the system. This is necessary in order to test the flexibility of the system, i.e. its ability to cope with a range of differing user demands, its ability to provide content tailored to each visitor's needs, i.e. their preferred reading language, age and interests, and the systems usability by people both with and without familiarity with the web.

Acknowledgements

This work was carried out as part of the EPSRC funded GUIDE project (GR/L05280) in cooperation with Lancaster City Council and has received support from HP Labs, Bristol and Lucent Technologies.

References

[1] Davies N, Mitchell K, Cheverst K, Blair G. Developing a context sensitive tourist guide. Proceedings of the First Workshop on HCI and Mobile Devices. UK: Glasgow, 1998.

[2] Davies N, Cheverst K, Mitchell K, Friday A. Caches in the air: disseminating information in the guide system. Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99). New Orleans, US: IEEE Press, 1999. p. 11–9.

[3] Cheverst K, Davies N, Mitchell K, Friday A. The role of connectivity in supporting context-sensitive applications, Handheld Ubiquitous Computing '99, Karlsruhe, Germany, 1999, submitted for publication.

[4] Schilit B, Adams N, Want R. Context-aware computing applications. Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, U.S., 1994.

[5] Brown PJ, Bovey JD, Chen X. Context-aware applications: from the laboratory to the market place. IEEE Personal Communications 1997;97:58–64.

[6] Long S, Kooper R et al. Rapid prototyping of mobile context-aware applications: the cyberguide case study. Proceedings of the Second ACM International Conference on Mobile Computing (MOBICOM), Rye, New York, U.S., 1996.

[7] Acharya S, Alonso R, Franklin M, ZDonik S. Broadcast disks: data management for asymmetric communication environments. Proceedings SIGMOD '95, San Jose, California, 1994. p. 199–210.

[8] Acharya S, Franklin M, Zdonik S. Balancing push and pull for data broadcast. Proceedings of the ACM SIGMOG '97, Tuscon, Arizona, 1997.

[9] Fujitsu TeamPad Technical Page. http://www.fjicl.com/TeamPad/teampad76.htm, 1999.

[10] 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, 1997.