Design of an Object Model for a Context Sensitive Tourist GUIDE

Keith Cheverst, Keith Mitchell and Nigel Davies
Distributed Multimedia Research Group,
Department of Computing,
Lancaster University,
Lancaster, LA1 4YR.
e-mail: kc,mitchelk,nigel@comp.lancs.ac.uk
telephone: +44 (0)1524 594337

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 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.

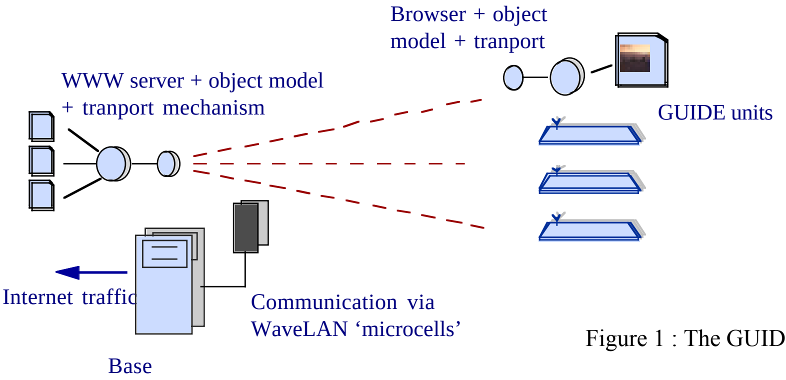

Figure 1 : The GUIDE Architecture.

The support for context-sensitive information by GUIDE has been strongly influenced by the work of Schilit [Schilit,94] and Brown [Brown,97]. 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 schedule should be used for transferring information to the portables units. Each base station being responsible for broadcasting information 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. Figure 1, below, illustrates this basic GUIDE architecture and shows how the GUIDE system will utilise web based technologies in order to disseminate information to mobile GUIDE units.

An alternative to the network based approach, would have been to produce some form of stand-alone system in which the entire working set of information is stored on the end system. Such an approach was adopted by the Cyberguide project [Long,96]. 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 unit's 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. 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 PC's, will consume less power and be cheaper.

In this paper we present the results of our requirements analysis and early 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.

[1] Requirements for an Intelligent Visitor Guide

Following a series of semi-structured interviews with members of Lancaster's Tourist Information Centre (TIC) we have obtained the following set of requirements for an intelligent visitor guide :-

Flexibility

Visitors should be able to use the GUIDE system as much or as little as they desire. For example, if a visitor requires a structured guide of the city then the system should be capable of directing them from one location to the next, providing relevant information throughout the tour. Alternatively, if a visitor would prefer to be left to roam the city, but still have the ability to ask the system for specific pieces of information along the way, then this mode of operation should also be supported.

Context-Sensitive Information

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 personal 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.

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 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.

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. The GUIDE system should provide support for such interactive services.

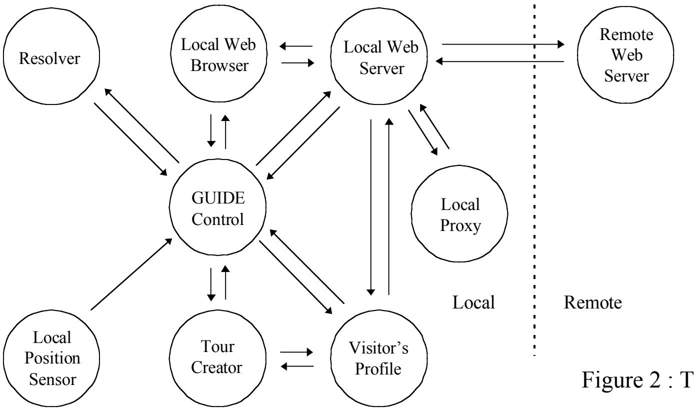

Figure 2 : The GUIDE object

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 that 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, 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 figure 2. It is worth noting that all components of the object model are Java based except for the local web browser component, which was written using C++ and 'Active X' technology.

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 visitor's 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 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 of the visitor’s new location e.g. the castle.

3) The control object requests an instance of the appropriate landmark object (e.g. the castle object) from the resolver 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.

4) The resolver object checks whether an instance of the appropriate landmark object already exists. If the instance exists in local memory, (i.e. because the visitor has already visited the location that day) then this instance is returned. Alternatively, if the instance only exists in the local cache, (i.e. because the visitor has visited this location in the past) then this instance is returned. However, if the instance does not reside at either of the above object stores, then an instance of the required object is created based on the visitor’s preferences and the current context

5) On receiving a handle to the landmark object, the control object can invoke methods on it. 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.

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 served by the local web server object which would obtain the required packets of 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.

4. Supporting Infrastructure

We have 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"x9"x1.5", 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 at [Fujitsu, 98].

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

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

Flexibility of Services: The system should support 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 disrupt the services provided by GUIDE as visitors roam around the city.

To meet these requirements we propose to replace TCP as the means of communication between the mobile units (clients) and base stations (servers). In more detail, the protocol will build on previous work on broadcast disks as a means of disseminating data [Acharya,95], [Acharya,97] to allow the system to support a large number of clients within each network cell whilst making use of the available bandwidth.

5. 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 is currently approaching the stage where the system can be demonstrated and evaluated by end-users.

REFERENCES

[Acharya, 94] Acharya, S., R. Alonso, M. Franklin, and S. ZDonik. "Broadcast Disks: Data Management for Asymmetric Communication Environments", Proc. SIGMOD '95, San Jose, California, Pages 199-210.

[Acharya, 97] Acharya, S., M. Franklin, and S. Zdonik. "Balancing Push and Pull for Data Broadcast", *Proc.* ACM SIGMOG '97*,* Tuscon, Arizona.

[Brown, 97] Brown, P.J., J.D. Bovey, and X. Chen. "Context-aware applications: from the laboratory to the market place", IEEE Personal Communications, October '97, Pages 58-64.

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

[Long, 96] Long, S., R. Kooper et al . "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.

[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.