Hermes: Generic Designs for Mobile, Context-Aware Trails-Based Applications

Cormac Driver Siobhán Clarke

Distributed Systems Group,
Computer Science Department
Trinity College Dublin,
Ireland
{Cormac.Driver, Siobhan.Clarke}@cs.tcd.ie

Abstract. The growth of ubiquitous computing has given rise to a range of possibilities for context-based application development. Technologies for mobile computing and context-awareness can be combined to create innovative ubiquitous computing applications. However, the ubiquitous computing paradigm poses a number of difficult challenges that must be repeatedly faced by developers. For each ubiquitous computing application developed, issues such as the unreliability of network connections, how to reason about context data, and how to provide easily usable interfaces must be dealt with by developers. This paper addresses a generic framework to support the design and implementation of mobile, context-aware applications that are user-activity-centric, and consider the core concept of a *trail*. A trail is a collection of locations, together with associated information and activities, and a dynamically reconfigurable recommended visiting order. We consider mobile, context-aware trails-based applications to be archetypal ubiquitous computing applications and therefore good examples on which to base the development of a framework.

1 Introduction

The development of ubiquitous computing applications poses numerous challenges to software developers. Issues inherent to the ubiquitous computing paradigm must be tackled during each application development effort, meaning that developers repeatedly encounter the same or similar issues, regardless of the application under consideration. These issues range from low-level programming issues to high-level usability issues. Notable examples of such issues include:

- Intermittent network connectivity must be catered for by making provisions for disconnected operation.
- Context-based application adaptation requires novel algorithms for context-based inference if the adaptation behaviour is to reside on resource constrained mobile devices.
- Providing an intuitive representation of the user's environment requires significant research into usability for ubiquitous computing applications.

Hermes is a software framework for mobile, context-aware trails-based applications which will support developers by providing generic components containing structure and behaviour common to all trails-based applications. At the most general level, a trail can be thought of as a collection of locations, together with associated information and activities, and a dynamically reconfigurable recommended visiting order. The trail is a collection of connected locations rather than a strict sequence since it may contain alternative sub-routes to cater for such variables as different modes of transport or other user preferences. Trails underpin a wide range of useful applications for a mobile user who has a set of activities that may or should be carried out throughout the day at different locations. Combining the trails concept with mobile, context-aware technology creates opportunities for innovative activity-based application development. Mobile, context-aware applications are those that run on wireless devices e.g., PDAs, and have an awareness of the physical and social situation in which they are deployed. Examples of trails applications that are both mobile and context-aware include tour guides, courier support/management systems, basic route planners, treasure hunt games and student support systems.

The Hermes framework will facilitate the development of a diverse range of realistic trails applications. This research is relevant to the field of ubiquitous application development as a whole because we believe trails applications to be archetypal ubiquitous computing applications. Mobile, context-aware trails-based applications exhibit the full range of ubiquitous computing characteristics as described in [1] and listed below:

- impact on social environment (privacy, perceptual pervasive observation, HCI, CSCW)
- task dynamism and task support
- device characteristics (devices that users interact with, embedded devices)

The major research challenges in developing the Hermes framework concern a) capturing trails application components in a generic manner for reuse and extension in future trails applications b) providing trail dynamic reconfiguration capabilities to ensure the provision of a more consistently realistic representation of the user's environment c) managing the trade-off between using resource-limited mobile devices with intermittent network connectivity and the significant application processing requirements of the trail generation and dynamic reconfiguration behaviour and d) discovering application usability models and providing for the development of applications that adhere to them.

The remainder of this document is as follows. Section two contains a brief state of the art in the area of mobile, context-aware trails-based applications. Section three contains a description of our work to date on Hermes. Section four contains a summary.

2 State of the Art

The mobile, context-aware trails-based applications that have been implemented to date are not diverse in terms of target audience or sophisticated in terms of ability to accurately reflect the user's environment. Tourist guide applications featuring trail generation and basic reconfiguration capabilities represent the current state of the art.

The GUIDE system [2], a mobile, context-aware trails-based tourist guide application, is the most advanced research effort in the field. The system contains a facility to request a structured tour based on a set of attractions that the user wishes to visit in the city of Lancaster, UK. Following tour generation the visitor is presented with a recommended sequence for visiting the chosen attractions. The ordering of the tour can dynamically change while the user is following it e.g., if the user stays longer than anticipated at a certain attraction. In relation to the Hermes project the GUIDE system has two main drawbacks. Firstly, the dynamic trail reconfiguration behaviour is quite basic, only occurring "periodically". Only positional and temporal context are sensed automatically, giving rise to the possibility of active trails being based on stale information as the remaining context must be manually uploaded. Secondly, in terms of software, the system was designed from the ground up with only the GUIDE tourist application in mind. This means that the opportunity to reuse GUIDE components in future development efforts is limited. The GUIDE system also encountered the challenges posed by ubiquitous computing issues such as the unreliability of wireless communications, application usability, and how much decision making power to give the application. The GUIDE developers are currently researching solutions for the difficulties they encountered.

The Cyberguide project [3] involved the development of two basic context-aware tour guide protoctypes and experienced similar issues to those encountered on the GUIDE project. The Stick-e Document [4], essentially an electronic post-it note, can be used to build basic mobile, context-aware tour guide applications. However, implementing a dynamically reconfigurable tour using Stick-e technology requires the specification of a note for every conceivable tour situation (user/context combination) before the application is deployed. A mobile aquarium tour guide system is described in [5]. This system features trail creation functionality but does not automatically sense context.

3 Hermes

"Frameworks are reusable designs of all or part of a software system described by a set of abstract classes and the way instances of those classes collaborate" [6]. Framework development is an iterative process driven by multiple application development efforts, with framework refinements following the development of each application. The result of this process is a generic framework that can cater for the requirements of many different applications within a specific domain. The approach taken on the Hermes project adheres to this method, implementing a series of mobile, context-aware trails-based applications and refining the framework as appropriate. Below we describe our work to date.

Analysis

We began the framework development process by specifying requirements for four dissimilar mobile, context-aware trails-based applications. These were a city route planner, a courier support system, a team-based treasure hunt game and a student support system. The route planner application allows users to follow city trails consisting of at least two locations – a start location and a destination location. The trails are affected by both personal and environmental context. The courier support system is used by mobile delivery couriers to manage their delivery schedule and to navi gate between pickup and delivery locations. The cour-

	Navigation	Context-Awareness	Trail Adaptability	Compulsory Activities	Optional Activities	Group Communication
Route Planner	✓	✓	✓	-	✓	-
Courier Support	✓	✓	✓	✓	-	✓
Treasure Hunt	✓	✓	✓	✓	✓	✓
Student Support	✓	✓	✓	✓	✓	-

Table 1. Analysis of mobile, context-aware trails-based applications

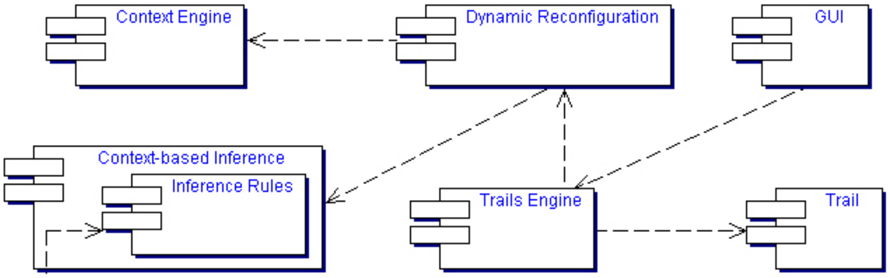

Fig. 1. Component diagram of initial framework design

ier's activities are compulsory and context information is used to find the quickest paths possible and to detect trail deviation. The treasure hunt application is a team-based game played by mobile players looking for a series of clues located throughout a ubiquitous computing environment. Trails are used to navigate between clue locations. The student support system allows new students at Trinity College Dublin to generate campus-wide trails including both compulsory and optional activities. The trails used in each of the applications are automatically dynamically reconfigured based on changes in context information.

Using these requirements we constructed design diagrams (UML) for each application. An examination of the requirements documents and design diagrams exposed the commonality that exists between the applications. Table 1 presents the results of the application analysis. The first column lists the applications examined. The first row contains a list of application features which we believe to be representative of what a generic mobile, context-aware trails-based application should support. These are described below:

- Navigation: the ability to direct the user from one location to another. The directions presented may not be as simple as the shortest route as they are influenced by the context.
- Context-Awareness: having an awareness of the physical and social situation in which the application is deployed. Context information used by the applications considered includes both personal and environmental data e.g., location, time, activity time constraints and priorities, weather, user preferences, awareness of other application users, and historical trail data.
- Trail Adaptability: the ability of a user's active trail to dynamically reconfigure based on changes in context.
- Compulsory Activities: those activities on a trail that must be completed. Such activities place constraints on the trail possibilities e.g., the courier must be in certain locations at certain times.
- Optional Activities: those activities on a trail that must not necessarily be completed by the user e.g., a student who is exploring the Trinity College campus in her free time does not have to complete every activity on her trail.
- Group Communication: the ability of application users to communicate reliably with each other.

Our analysis illustrates that although the set of trails-based applications have significantly different high-level requirements, the high level of commonality that exists between their requirements for trails-related functionality renders framework support valuable.

Initial Framework Design

Figure 1 shows a UML component diagram containing the components found to be common across the four applications designed in the first phase of our work. The UML class diagrams for each of the applications were analysed for commonality and cohesive elements were grouped into components.

The Context Engine component is responsible for providing context data. This involves sourcing relevant context data from numerous distributed sources and rendering and distributing it in a manner appropriate to the application under consideration.

The Trails Engine component is responsible for managing the lifecycle of a trail. Trails are created based on input data from both the user and the Context Engine. Following this, the component is responsible for managing the trail as the user embarks on it, ensuring the trail's relevance at all times.

The Trail component provides a software representation of a trail that is created and followed by the user. This component is manipulated by the Trails Engine component.

The Dynamic Reconfiguration component is responsible for monitoring environmental context throughout the active lifetime of a trail and taking the appropriate actions to ensure that the trail consistently reflects the user's environment. The component uses context data to provide the Context-based Inference component with information and then uses the algorithms for context-based inference contained in this component to arrive at decisions regarding active trail reconfiguration. The Dynamic Reconfiguration component then advises the Trails Engine component regarding appropriate trail alterations.

The Context-based Inference component, which contains an Inference Rules subcomponent, is responsible for the aggregation and interpretation of context data. Context data, received from the Dynamic Reconfiguration component, is analysed with a view to discovering which actions need to be taken in order to keep the active trail in a state that accurately reflects the environment in which the user exists. The Inference Rules subcomponent contains rules to assess real-time context data.

The GUI component is responsible for presenting the trail application to the user and providing a way for the user to communicate with the underlying application and vice versa.

3.1 Early Framework Implementation

In order to build the first version of the Hermes framework we are implementing a student support application based on mobile, context-aware trails-based technology. The application will be used by new students at Trinity College Dublin during their initial period at the college and caters for the generation of campus-wide trails including both compulsory tasks such as registration and optional activities such as visiting college buildings and joining societies. The trails are both based on and affected by environmental and personal context, and are dynamically reconfigurable.

This first version of the framework will be further refined and developed into a generic framework through the implementation of the remaining three applications.

3.2 Usability

A key factor in the success of any application is usability. Early research on task analysis and more recent work on contextual, collaborative and social factors has informed our approach to the design of user interfaces. However, the end-user devices e.g., PDAs, used in ubiquitous computing environments pose new usability challenges related to limited interface capabilities and removal of the mouse and keyboard as the main input devices. Investigation is required to assess the ways in which these changes affect the manner in which end-users interact with PDA-based applications. Additionally, there is a greater variety of environments and contexts in which an application may be used, with wireless connectivity allowing mobile users to receive information on the move. This situation introduces a range of information presentation issues e.g., what to display and when, that must be dealt with if sufficiently usable applications are to be developed.

Existing usability studies within the mobile, context-aware trails-based application field have tended to centre on specific systems, rather than classes of system [2, 5]. This presents us with an opportunity for research that will impact on the usability of a whole genre of ubiquitous computing applications, namely trails-based applications, which exhibit the essential features of ubiquitous computing applications. Our aim in this area of the Hermes project is to investigate the usability of a range of trails-based applications, concentrating on specific, problematic features in order to produce design guidelines, and produce an approach to evaluation and set of evaluation criteria appropriate for trails-based applications.

Application Evaluation

The student support system will be used and evaluated by two classes of M.Sc. students in Trinity College Dublin's Computer Science Department who begin their program of study in October 2004. On average there are twenty students in an M.Sc. class and each student (subject to hardware constraints) will use the application to follow campus-based trails consisting of compulsory activities such as registration and introductory lecture attendance, as well as optional college exploration activities.

Among the possible techniques we are considering for observation of students include video, note-taking, traditional lab-based usability testing, lag sequential analysis, and intensive interviewing. As well as the use of note-taking, video, and audio recordings, data can also be collected through logging within the software. Recording of all screen activity (a common technique for desktop systems), while still useful, is less informative on handhelds due to the lack of a mouse pointer, and the greater variety of contextual factors.

4 Summary

The evolution of ubiquitous computing has resulted in a wealth of opportunities for innovative application development. However, the existence of issues inherent to the ubiquitous computing paradigm requires developers to repeatedly tackle the same set of difficult challenges across distinct development efforts. Through the design, implementation and evaluation of a series of mobile, context-aware trails-based applications we are producing a framework to support to development of these applications, hence relieving developers from continually dealing with the same set of issues. The trails model was chosen as it exhibits the major characteristics of ubiquitous computing applications.

References

[1] G. Banavar, A. Bernstein. "Challenges in Design and Software Infrastructure for Ubiquitous Computing Applications". Advances in Computers, Vol. 62, 2004.

[2] K. Cheverst, N. Davies, K. Mitchell and A. Friday. "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project". In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, p. 20-31, Boston, Massachusetts, 2000. ACM Press.

[3] G. Abowd et al. "Cyberguide: A Mobile Context-Aware Tour Guide". ACM Wireless Networks - Special issue: Mobile Computing and Networking: selected papers from MobiCom '96, p. 421-433. New York, 1997.

[4] P.J. Brown. "The Stick-e Document: a Framework for Creating Context-aware Applications". In Proceedings of International Conference on Electronic Documents, Document Manipulation, and Document Dissemination (EP 96), Palo Alto, California., 1996, pp. 259-272. Proceedings published in Origination, Dissemination, and Design (EP-ODD), vol. 8, no. 1, John Wiley & Sons, New York, 1996, pp. 1-13.

[5] F. Bellotti, R. Berta, A. De Gloria and M. Margarone. "User Testing a Hypermedia Tour Guide". IEEE Pervasive Computing, Vol. 1, No. 2, April-June 2002.

[6] D. Roberts, R. Johnson. "Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks". ACM Pattern Languages of Program Design 3, 1997, pp. 471-486.