Sharing Context History in Mobile, Context-Aware Trails-Based Applications

Mike Spence, Cormac Driver, Siobhán Clarke
Department of Computer Science
Trinity College Dublin
Dublin 2, Ireland
{spencem, driverc, sclarke}@cs.tcd.ie

ABSTRACT

The growth of ubiquitous computing has given rise to a range of possibilities for context-based application development. The Hermes project is addressing the development of a generic framework to support the design and implementation of mobile, context-aware applications by focusing on the core abstraction of a trail. This paper discusses a major element of our current work - augmenting the Hermes framework with collaborative context and context history components. These components will give application developers the ability to obtain, manage, and exploit current and historical context information, such as trail histories, that can only be acquired via collaboration between a number of dedicated sensors and devices in the ubiquitous computing space.

Keywords

Trails, collaborative context, context history, mobile devices, context-aware, ubiquitous computing, framework development

INTRODUCTION

The development of ubiquitous computing applications poses numerous challenges to software developers. Many 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 and executing resource-intensive application adaptation algo-rithms on resource-constrained devices.

Hermes (http://hermes.dsg.cs.tcd.ie) is a software framework for mobile, context-aware trails-based applications that will support developers by providing generic components containing structure and behaviour common to all trails-based applications [2]. Mobile, context-aware applications are those that run on mobile devices, such as PDAs or smartphones, and have an awareness of the physical and social situation in which they are deployed. *Context* is defined as any information that can be used to characterize this situation [3]. A *trail* is a collection of selected activities, together with associated locations and other information, and a dynamically reconfigurable recommended visiting order.

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. Examples of trails-based applications that are both mobile and context-aware include tour guides, courier support/management systems, basic route planners, treasure hunt games and student activity support systems.

This paper discusses *collaborative context* - the sharing of context data between devices in a ubiquitous computing environment and *trail histories* – a recorded account of how a user makes use of a trail. We have recently begun work on adding collaborative context and context history comp-onents to the existing Hermes framework. The remainder of this paper contains an overview of the collaborative context concept and its challenges; a discussion of trails, trail histories, and their challenges; and concludes with a presentation of planned future work and open questions.

COLLABORATIVE CONTEXT

Fundamental to ubiquitous computing applications is a requirement to sense or obtain relevant information from the user's environment. In a mobile, context-aware trails-based situation, that environment can be spread over a relatively large geographical space e.g., a city, and the types of context information required include those not directly provided by typical dedicated sensors e.g., crowd density at distant locations on a user's trail and ratings of activities by users. For this paper, we differentiate *dedicated sensors*, such as accelerometers, thermometers, or GPS sensors, from the higher-level *devices*. Dedicated sensors are built to sense a single aspect of the environment and communicate that value. A device might have a sensor embedded in it, but the property of being able to combine that sensor information with other information makes it a device. Devices include servers, laptops, mobile devices, and other computing platforms that are not dedicated sensors.

Collaborative context is the subset of context that is acquired via the communication between dedicated sensors and devices in a ubiquitous computing space and can be used to increase both the range of context available and context reliability. More formally, collaborative context is defined recursively as:

- Context information acquired directly from other devices that are not dedicated sensors
- Context information acquired from dedicated sensors through other devices
- Context information derived from the combination of *collaborative context* and possibly sensor data and/or context information already on the device

Context obtained directly from dedicated sensors is not included in collaborative context, because another device does not have access to this context; this means another device *cannot act on the context data before it is received*. This property of collaborative context is the important element in the challenges presented in the next section

An example of using collaborative context in an application is sharing trail histories to revise activity or trail details. A trail history is a recorded account of how the user makes use of a given trail. This includes the initial application-generated trail, any reconfigurations of that trail, the trail actually followed by the user, and other context information that might aid in explaining why the user might have deviated from or followed the trail. If an application on a user's device receives this historical context information from other devices, it can infer from it, for example, that a particular activity took longer than originally expected. The application might then lengthen the predicted duration of that activity. This form of collaborative technique can be used to provide a more realistic trails experience to mobile users.

Collaborative Context Challenges

The use of collaborative context has the potential to greatly improve both the manner in which trails are generated and the representation of the ubiquitous computing space to the user; however, there are a number of significant research challenges in this area. Rather than concentrating on the communication challenges for requesting and transferring context between devices, our research in collaborative context focuses on how context information is disclosed and how context information is refined upon acquiring it. Consequently, these are the three challenges of collaborative context that we are addressing:

Privacy

Privacy addresses the conditions under which an application stores or discloses a piece of information. Some applications will not function without being provided some form of context information, even if it is less precise e.g., disclosing city or country instead of exact GPS coordinates. As discussed in [5], most users will only agree to provide a system with personal data if the benefits gained from the disclosure outweigh the risks e.g., purchasing items by credit card on the Internet.

Trust

Trust is the amount of confidence a party has that another party will behave as expected. Trust in relation to collaborative context is a function of the confidence in the original source of the context information, as well as any other device it passes through en route to the destination. It can also vary based on the type of the context information.

Perception

The term perception is used to describe the difficulties of relating a piece of context from an outside environment into the device's environment. In collaborative context, the originators or intermediary manipulators of context information are normally in a different situation than the receivers of that context. Whether this difference is that of location, application goals, or a combination of factors, the context a device communicates depends on its perception of the surrounding environment. Unfortunately, defining and communicating the entire relevant context is impossible for most applications. Imprecision, staleness, conflicting, incor-rect, or missing values, and subjective opinions and recommendations are all important perception issues.

To some extent, these are important challenges of acquiring and using any type of context; however, due to the indirect path and possibly subjective nature of the pieces of context, handling these issues is essential for collaborative context.

TRAILS AND TRAIL HISTORIES

Current context is not the only source of useful collaborative context to an application; historical context can also be an important source of shared context. A key component of collaborative context in trails-based applications is the ability to share past and current trails. Some benefits gained by the ability to use and share trails and trail histories are:

Avoiding trail generation

Due to the limited processing power of the mobile devices and the processing-intensive nature of trail generation, using a previously recorded trail to avoid calculating the entire trail again can have beneficial effects on trail quality and application responsiveness.

Verifying assumptions

Trail histories can be used as an aid in verifying or revising certain assumptions made by the application about the activities and the trail as a whole, such as activity duration or journey time between activities.

Utilizing feedback for adaptation

Trail generation utilizes user preference and other context information to predict in what order and which route users will want to experience a trail. Trails are unique to most predictive context-aware applications in that there is implicit feedback on whether the user is satisfied with the current generated trail i.e., if the user follows the trail. This feedback can aid in predicting under what conditions a user might deviate from a particular trail, and an application can use that information to generate a more acceptable trail.

Synchronizing users

Trails and trail histories may be shared among friends and coworkers to allow users to synchronize their activities and routes between activities. Current trails can be used to determine where a user plans to be, and trail histories can be used to predict where a user might be.

Offloading trail computation

Another device might have greater knowledge of the activities or routes between the activities. In order to leverage the knowledge of that device, part of a trail or a collection of activities can be given to that device to order the activities more appropriately or choose more suitable routes. This is also an indirect method of using information contained in another device's trail histories without the overhead of actually passing the trail histories themselves. Additionally, if groups of activities can be partitioned for ordering like this, the runtime complexity of the trail generation can be significantly reduced; however, finding partitions generically is a nontrivial task [4].

Trails and Trail Histories Challenges

Sharing historical context creates challenges, because the information is not current and the interaction occurred in another device with a different user context. Consequently, even though sharing this information has important privacy and trust issues, representing and utilizing trail histories deal mostly with the perception challenge.

Challenges in representing trail histories

The choice of what context information to store in the representation of a trail history is vital for trail reuse, as this additional context might help explain why an application generated a particular trail or why a user might have deviated from it. If the application that created the trail history includes a useful subset of motivating context information, this removes the trail history exploiter, which might also be the trail history creator, from attempting to reconstruct that information itself. However, discovering an optimal subset of available context information to store in a trail history can be difficult as the most useful subset can be specific to the exploiting application's user, device, and situation.

Trail history size is also an important representation concern when considering the limited storage space and low-bandwidth communications on mobile devices. Consequently, finding the optimal subset of the available context information to store in a trail history is also crucial for use by resource-constrained devices. To alleviate the size concerns, an application could store or pass only a subset of the trail history; however, determining the correct subset for later, possibly off-device, reuse can be difficult, and this still involves the issues associated with finding a subset of additional context to include.

Dynamic trail reconfiguration presents a final trail history representation challenge. As a user follows a trail, minor adjustments such as an activity running a few minutes over or a route between activities not taking as long to traverse as expected might be useful to store. More dramatic adjustments such as an added activity or a route closure might force the activity ordering to change. Finding a way to represent these changes in way that is efficient for applications to examine and is sensitive to the resource-constrained nature of mobile devices is essential.

Challenges in utilizing trails and trail histories

Other challenges to actually utilizing generic trails and trail histories, rather than just representing them, are:

- Analyzing trails: Determining what portions of the trail are relevant can be difficult, particularly when they contain activities unfamiliar to an application.
- Comparing trails: Different trails probably have different orderings based on the users' preferences and environment conditions at the time of creation and use. They might also contain different activities.
- Merging trails: This is closely related to analyzing and comparing trails, which are necessary components for merging. While merging multiple trails into one might prevent an application from recalculating an entire trail, it still must find a way to mix and connect the activities in the various trails. There is also a danger that merging might be more expensive than simply generating the trail.

Like most user interactions with their environment in ubiquitous computing, the underlying issue in most of these challenges is that trails and trail histories are complicated pieces of derived context and are influenced by a combination of several factors that are difficult to capture and reuse.

CURRENT WORK AND OPEN ISSUES

A trails-based multi-player riddle game based in Dublin city is currently under development. The object of the game is to collect more points than the other players by the end of the game. A player receives points by answering riddles at a collection of locations and a correct answer removes the riddle at that location. There are a limited number of riddles at each location, all with varying types and point values. The attempts to answer riddles are the activities in the trail and the trail generation and reconfiguration are based on current game state and player-defined strategy preferences.

This application will be used as a platform upon which to design and implement collaborative context and context history components that will aid the enhancement of existing Hermes framework components. The collaborative context component will be used to pass context information such as game state and trail histories.

Most of the applications of trail histories to the riddle game will be to aid the generation of trails and improve game strategies. Part of our evaluation will be for users to play the game multiple times. Both the user's and other successful players' trail histories from previous games can be analyzed and exploited to create more successful, customized player trails.

Game simplifications, like a fixed collection of previously defined riddle activities, limited geographic space, and a fixed time limit should allow us to focus on a subset of the difficult challenges presented earlier, such as game-specific trail history representation and exploitation on constrained-resource devices.

Allowing a user to play multiple times will also help us answer a key question concerning mobile, trails-based applications: While studies such as the GUIDE project [1] have shown that a one-time use of a trails-based application in an unfamiliar environment can be a great aid to users, once the users are familiar with the geographic area and the activities, will the trails-based application be useful? Our initial feeling is that this will depend on the quality of the context information acquired and how an application utilizes it to aid users in their tasks. If players continue to use the trails-based application features even after becoming familiar with the area and activities, this will certainly be a positive step in demonstrating the utility of trails-based applications.

ACKNOWLEDGMENTS

Mike Spence is funded by Intel Corporation. Cormac Driver is supported by the IRCSET Embark Initiative.

REFERENCES

1. Cheverst, K. et al. "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project", in *Proceedings of sixth ACM International Conference Mobile Computing and Networking (MOBICOM'2000)*, Boston, U.S., ACM Press, pp. 20-31, August 2000.
2. Clarke, S. and Driver, C. "Context-Aware Trails". *IEEE Computer*, Vol. 37, No. 8. pp. 97-99, August 2004.
3. Dey A. and Abowd G., "Towards a Better Understanding of Context and Context-Awareness", *Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness,* The Hague, Netherlands, April 1-6, 2000.
4. Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.
5. Schilit B. et al. "Ubiquitous location-aware computing and the 'Place Lab' Initiative". *In Proceedings of the First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspot*. San Diego, CA, Sept. 2003.

Author Biographies

Mike Spence received his B.A. in Computer Science from Rice University in 2000. After working for Scient Corporation and Closest Point Solutions as a developer, he returned to school and received a Postgraduate Diploma in Ubiquitous Computing from Trinity College Dublin. He is currently a PhD student in the Distributed Systems Group at Trinity College Dublin under his supervisor, Siobhán Clarke.

Cormac Driver holds a B.Sc. in Business Information Systems from the Dublin Institute of Technology and a M.Sc. in Networks and Distributed Systems from Trinity College Dublin. He is currently a PhD student in the Distributed Systems Group at Trinity under Siobhán Clarke and is working on the Hermes project.

Siobhán Clarke is a lecturer in the Department of Computer Science at Trinity College Dublin. Her research interests are in design and programming models for mobile, context-aware computing.