CONTEXT DEPENDENT SERVICES IN AN M-LEARNING ENVIRONMENT: THE PRINTING CASE

Anna Trifonova
Dipartimento di Informatica e Telecomunicazioni
Università degli Studi di Trento, 38050 Povo (Trento) Italy
trifonova@science.unitn.it

Marco Ronchetti
Dipartimento di Informatica e Telecomunicazioni
Università degli Studi di Trento, 38050 Povo (Trento) Italy
Marco.Ronchetti@unitn.it

ABSTRACT

We study a context-dependent problem that arises in an m-learning environment: how to print a document when being in a generic place. We implement a working solution, and argue about the general value of our approach. We examine the printing problem because it has a practical value, and at the same time it turns out to be a prototype for a class of other problems that can be encountered when moving from e-learning to m-learning.

KEYWORDS

m-learning, printing, location-dependent services

1. INTRODUCTION

"m-learning" is one of the successful buzzwords of the beginning of the millennium. It combines the promises of two very promising fields: e-learning and mobile computing. E-learning is growing at a very fast rate: nowadays most universities have at least some degree of support for e-learning, companies are investing in the field, and the need for continuous education pushes for e-learning solutions. On the other hand it is likely that mobile telecommunication will continue to grow and to add new services. Competing and complementary wireless technologies like wireless LAN, Bluetooth, GPRS and UMTS will multiply potential handheld applications. IDC forecasts that 63 millions handhelds will be sold by 2004, and that approximately 38% of them will be smart phones, integrating PDA functionality with features for communication. Most mobile clients will support Java (J2ME) making it easier and less costly to develop portable applications. Given such scenario, forecasting the success of m-learning seems to be an easy bet. It is more difficult to understand in detail how m-learning will help reaching the goals of a better learning, and how it will be different from the rest of e-learning. According to the literature [Zobel 2001, Steinberger 2002] successful m-learning will be characterized by the following properties:

5 minute value: the ability to use small fragments of time (e.g. waiting time) for learning (e.g. doing quizzes, using a discussion forum, communicating, reading material);

Simplicity: the limited display and input capabilities of the mobile devices make it difficult to use rich (e.g. complex and multimedial) documents using a PDA-like interface: it is therefore not useful or practical to transpose a power-point presentation on a PDA;

Context dependent information: the dependency can be relative to:

- location context; i.e. the system knows the location where the learner resides and adjusts to it;
- temporal context; i.e. the system is aware of time dependent data;
- behavioral context; i.e. the system monitors the activities performed by the learner and responds to them adjusting its behavior;
- interest specific context: i.e. the system modifies its behavior according to the user's preferences.

Examples of context-dependent systems (although not related to m-learning) are:

oTourist information systems, like GUIDE [Cheverst 2000] and CYBERGUIDE [Abowd 1997]: these systems offer information to tourists, taking into account their current location.

oContext-aware messaging systems that trigger actions according to a specific context, like the ComMotion system [Marmasse 2000] which links personal information to locations and generates events (e.g. sound or message boxes), when a user moves to a certain location. Other such system are CyberMinder [Dey 2000] and Icron [Heidmann 2000]: they allow the user to define more complex conditions, like time-and-location dependent conditions.

oGeneral utilities, like "Friend finder", "GeoNotes", "BusLocator" [Nord 2002]

We experiment with such requirements, addressing a simple task that is often necessary in all e-learning environments, and that is in general taken for granted: the ability to print a document. This quick, simple task (if performed in a non-mobile environment) becomes less trivial when performed from a mobile device, like a palmtop/laptop equipped with a wireless-LAN card. We show that this apparently trivial task contains elements that constitute a template for other problems that can be experienced when approaching e-learning from the mobile side.

2. THE PROBLEM AND POSSIBLE SOLUTIONS

"Printing on the nearest suitable printer" contains elements that can be found in many other mobility-related problems. Let's first examine the traditional printing process: printing happens from any application, and uses services that are provided by the (local) operating system (OS). The user typically interacts with the OS through some API to choose his/her favorite printer. In general, a default printer is chosen once, and kept until some special need arise. If the printer is on a network, the OS is in charge of managing the network interface so as to be able to talk to the printer and pass the task to it.

If the computer is mobile, then we need to adapt to the location and to other conditions: such adaptation needs to be based on some space-time-dependent data, and might be performed on a local basis, or on an external resource (a server). So we add at least two functional elements: the discovery of time and location, and the decision of what to do on the base of the contextual data, possibly accessing also the user's preferences. Although such process might also involve behavioral data, we will not discuss in great detail this aspect.

Let's state the problem in some more detail. When a user wants to print from her mobile device, the system needs to find which is the most convenient accessible printer, inform the user about the choice, optionally allow her to modify this choice, and finally print the document, optionally allowing the user to monitor the status of the printer queue. (Note that by changing the words "print" with "use a service", and "printer" with "service provider" one maps this problem to a general one, and therefore this problem is prototypical for a class of other location-dependent problems).

"Finding the most convenient accessible printer" means performing two steps. The first is to gather information about location, time, behavior and preferences. The second is to choose the resource that best suites to needs.

Location is important because one needs to find the nearby printers. Time might limit the availability of some printers (some room might be locked at night or during week-ends). Behavior, i.e. knowing what the user is currently doing, might require a black and white printer if the user is reading a text-only document, or a color printer if the user is looking at pictures. Preferences might involve opting by default for cheaper services at expenses of print quality, or vice versa, etc.

Once this info is available, one needs to use it to find the resource. Therefore, the characteristics of all printers must be known, and matched against the requirements posed by the context. In principle, all the info about the printers might be kept on the mobile device, at least if mobility is limited, and if data are not so massive that they endanger the availability of the device's precious memory. One might also imagine that when moving to a different environment (e.g. to another building) the mobile device could discard all the info regarding the previous environment, and download the info relative to the present surroundings. However, such approach might miss real time info, like the current load on a printer, producing poor choices (it is probably better to walk a few more steps to an empty printer, that quickly reaching a busy one). A second possibility is hence that info about the printers is kept on some server: the mobile client could contact

the server passing its own context info, and getting back the indication of the chosen printer. The main drawback of having a centralized server is the scalability of the solution, in terms of performance (the central server becomes a bottleneck), reliability (the server becomes a critical single point of failure) and geographic scale (it makes no sense of thinking of a central server that knows about all printers in town). One can overcome these weaknesses in a standard way, i.e. by having a federation of servers (each being responsible for a sub-region, and being able to forward requests to other servers) with some degree of replication.

Another issue has to be considered. In order to be able to perform a print, the mobile device needs to know about the printer, i.e. it must have a driver for it. This makes life complex, because we need to foresee in advance which printers will be used, that again is not practical. One could think of downloading on demand the needed drivers, but sometimes installing a driver requires rebooting the machine, so also this solution is not sensible. To avoid the problem, we might ask the server to print on our behalf. After all, this is what is typically done in a multi-user OS, where printers are never directly accessible by the users (to stay away from nasty concurrency problems), and has the advantage of enabling accounting and permission checking.

Asking the server to print can take two roads: either we pass to the server the current version of the document, and the information about the application that is using the document, or we pass a printable device-independent version of the document (such as a postscript file). The first option is again not convenient, since it requires the server to carry all possible programs, and to start them when necessary (further limiting the usability of servers running OS A (e.g. Linux) from machines using OS B (e.g. WindowsCE). The second is feasible: for instance it is possible to print Postscript files also on non-Postscript printers, e.g. using (on the server) the Ghostscript program that is available for different operating systems. All that is needed on the mobile side is a printer driver that produces a Postscript file: a requirement is easy to comply with.

At this point, what we call "printing" on the mobile device actually means "1-print the document to a postscript file", "2-pass to the server the context information and the generated file", "3-have the server choose the printer, send the postscript file to it, and pass back the info about the chosen printer". One last problem remains open: all this should happen when the user chooses the "print" menu item. This means that one should write a (pseudo) printer driver that, when invoked, performs all these actions. This is certainly possible, although it requires digging in OS-dependent technical details.

A less convenient but more straightforward solution is to explicitly perform the above sequence in two steps: first one prints to a postscript file, and then one explicitly asks the server to print it. A way to do it is through web pages: the mobile client contacts the server via HTTP through a browser. The server provides a form in which the user specifies the name of the postscript file. The form then triggers an active component (e.g. a servlet) that opens a socket to the client to retrieve the context-dependent info and the file, and after choosing the best printer prints the document and reports the choice in an HTML page that is returned to the mobile user. Of course, we have here implied that the mobile system is able to provide a service via socket to pass the context-dependent info. As an alternative, one could pass all this info through HTTP.

We implemented and tested this last solution, and we are working at the implementation of the most complete and convenient option, namely the one that is completely transparent to the user and that is based on the pseudo-printer driver.

As far as the positioning systems is concerned, we note that many possible solutions were provided in literature. Due to space constraint, we cannot discuss theme here: we only note that we believe that the most convenient systems are the ones that do not require additional hardware or infrastructure. We therefore experimented with a system [Battiti 2002] based on the strength of the IEEE811b signal coming from different antennas, that uses available hardware and infrastructure, and therefore requires only adding a software layer. In a conceivably wireless networked city, such method would work indoors and outdoors.

3. GENERALIZATION

We have seen that the printing problem can be mapped on a more general one, where the focus is on providing some context-dependent service, while using basic services provided by the infrastructure (e.g. by the OS, or by a Learning Management System).

The idea is to insert a software layer between the service requestor and the service provider. As we discussed, such software layer should in general refer to an external server for at least two reasons: the mobile component cannot be aware of all possible settings that are available in different places, and the optimal choice might depend by factors that could be dynamic, and therefore unknown by the mobile component. The external server must obtain context data from the requestor. At this point two choices are possible: either the server fully provides the customized service, or it provides a "meta-service", i.e. it only identifies the best option and then passes this information back to the requestor. The requestor then performs the actual customized service. In some cases (like in the printing problem) this last solution might be highly unpractical; in other cases however it might be a viable solution, and might even be preferable since it diminishes the workload on the server. As we have seen, implementing this middleware in a seamless way can require digging into technical details of the infrastructure (e.g. of the OS). In the particular case of the printing, it required writing (or at least modifying) a device driver, that is not a trivial task. In other cases, like for instance in the case of a service provided by a Learning Management System, it might mean entering in the (possibly proprietary) code of the infrastructure providing the service: a possibly prohibitive task. In such cases one can fall back to a less convenient, two step process: through the notion of a stub one would then use a local instance of the needed service (i.e. one might have an actor on the server that asks for a local service on behalf of a remote, mobile user).

4. CONCLUSION

We implemented a working solution to a context-dependent problem that arises in an m-learning environment: how to print a document from a mobile device when being in a generic location. Of course, this particular problem is non ***specific*** of the m-learning domain. However, besides having a practical value also in this particular domain, we believe it is relevant as being a ***prototype*** for a class of other problems that can be encountered when moving from a traditional environment to a mobile setting. We believe that the logical organization of the solution we propose can therefore also be transposed in such cases.

ACKNOWLEDGEMENT

The present work is related to the Wilma project financed by Provincia Autonoma di Trento. One of us (AT) is supported by the Centro Ricerche FIAT.

REFERENCES

Abowd, G. D. et al., 1997: Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3: 421-433

Battiti R., Brunato M., Villani A., 2002 "Statistical Learning Theory for Location Fingerprinting in Wireless LANs", private communication

Cheverst, K.; Davies, N.; Mitchell, K.; Friday, A.; Efstratiou, C., 2000: Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences, in Proceedings of CHI'00 (Netherlands, 2000), ACM Press

Dey, A., et al. 2000 CyberMinder: A Context-aware System for Supporting Reminders, Second International Symposion on Handheld and Ubiquitous Computing 2000 (HUC2K), Bristol (UK), Sept. 25-27, 2000, LNCS 1927, Springer, 187-199

Heidemann J. and Shah, D. 2000 Location-Aware Scheduling With Minimal Infrastructure. In USENIX Conference Proceedings, pp. 131-138. San Diego, CA, USC/Information Sciences Institute, USENIX. June, 2000. Available on-line at http://www.isi.edu/~johnh/PAPERS/Heidemann00b.html.

Marmasse, N.; Schmandt, C., 2000 Location-aware Information Delivery with ComMotion, Second International Symposion on Handheld and Ubiquitous Computing 2000 (HUC2K), Bristol (UK), Sept. 25-27, 2000, LNCS 1927, Springer, 157-171

Nord J., Synnes K., Parnes P., 2002 An Architecture for Location Aware Applications 35th Annual Hawaii International Conference on System Sciences (HICSS'02) January 07 - 10, 2002 Big Island, Hawaii, - Volume 9, p.293

Steinberger Claudia, 2002 "Wireless meets Wireline eLearning", in Proc. of ED-MEDIA 2002, June 2002, Denver (CO,USA)

Zobel J. (2001): Mobile Business and M-Commerce, Hanser.