UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SUPPORTING MOBILE LEARNING WITH LOCATION-AWARE PRINTING SERVICE

Marco Ronchetti and Anna Trifonova

March 2003

Technical Report # DIT-03-010

Supporting Mobile Learning with Location-Aware Printing Service

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

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

Abstract

Learning and studying are very often connected with reading printed materials. In this paper we describe some implementation issues of a common task – printing a document, when applied in a mobile environment. In a non-mobile context this task is simple and is usually taken for granted, but the introduction of a palmtop/laptop equipped with a wireless-LAN card makes the task less trivial. To fulfil the requirements placed by the mobility a system should consider some location-time-dependant data.

Keywords: mobile, context-aware, printing

1. Introduction

Mobile devices, such as laptops, PDAs and mobile phones are becoming more and more used not only by the business world people but also by the common people and even kids. The field of use of such devices is spreading from only personal organizers to wide variety of processes. It is becoming standard that the mobile device has an internet connection (i.e. WAP) or connection to local area network. This connection is usually wireless. Wireless LAN, Bluetooth, GPRS and UMTS together with the fact that soon most of the mobile clients will support Java (J2ME), thus making the application development easier, faster and less costly, will multiply the potential handheld applications. Recently lots of research is going on in the field of ubiquitous computing and location-aware applications. The research in location discovery in the local area network is giving more and more precise results (Battiti, Brunato and Villani 2002). On the other hand the fast-growing e-learning field encourages research in new forms of educational tools that include mobility and context as an essential requirement. "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. And the question is *how* m-learning will help reaching the goals of a better learning, and *how* it will be different from the rest of e-learning.

Mobile learning most probably applies best to processes, where specific knowledge should be retrieved/accessed in a certain moment, where discussions in distributed groups (i.e. brainstorming) appear, where data is collected or utilized "on the field", and where context-information is strongly related to the learning content. In literature there is indication that the success of m-learning will be most likely guaranteed if certain properties are carried out (Zobel 2001, Steinberger 2002, SLL 2001):

- Short, from 30 seconds to 10 minutes long modules. The participants should be able to use their small fragments of waiting time (i.e. waiting for a meeting or while travelling in a train) for learning, like reading small pieces of data, doing quizzes or using forums or chat for finding answers to "on field" questions.

- Simple, funny and added value functionality. The computational power and other properties (i.e. lack of input capabilities) of mobile devices make it difficult to use complex and multimedial content. It should be possible to use an m-learning system without having to read a thick user manual and one should find it more interesting or necessary and useful (or at least equally) to study using this m-learning system in his/her 5 min. break than playing a game on the same device.

- Area/Domain specific content delivered just in time/place and personalized interaction. 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) and interest specific context (i.e. the system modifies its behavior according to the user's preferences). Of course a mix of the contextual dependencies is possible and likely.

In this context we are addressing a simple task, which is necessary in most of computer-supported tasks, and in this sense, in all e-learning environments - printing. In a non-mobile environment this task is simple and is usually taken for granted, but in mobile situation, such as palmtop/laptop equipped with a wireless-LAN card, it becomes less trivial.

2. The problem and possible solutions

In few words the problem we experiment with can be expressed this way: "Printing on the nearest suitable printer".

Let's compare the non-mobile and mobile case of the printing process. In common one should be able to print from any application which actually uses the APIs of the operating system (OS). There is usually a default printer, local if exists, but more often the printer is on the network and the OS is managing the network interface and passes the job to the printer when needed. The OS has to be able to "talk" the particular printer's "language" in order to do the task correct (i.e. a specific printer's driver should be installed).

If the computer is mobile some more conditions should be added – location-time-dependant data should be considered. Based on this data and possibly to some other preferences, like user's access rights over a certain printer or pages limitations, a decision should be taken. This could be done locally or on an external resource – a server. The system should find which is the most convenient for this user and accessible in that moment printer; inform the user about the choice, possibly giving him the possibility to modify this choice. Finally the document should be printed, optionally allowing the user to monitor the status of the printer queue.

"Printing on the nearest suitable printer" contains elements that can be found in many other mobility-related problems. What makes the printing problem so prototypical, as we shall discuss in a later section, is the fact that printing is a service (and can be used from any application) and it uses services (those provided by the operating system). There are various architectural choices that can solve the printing problem. We shall briefly discuss them, outlining advantages and disadvantages of each of them.

The first step is collecting context information. In order to find the "closest" printer we need to know where the user/device is, i.e. their physical position. But then occurs the problem that the "nearby" printer could not be always reachable (some room might be locked at night or during week-ends). The behavioral information is also important. Knowing what the user is currently doing and what applications is using at that time could provoke the usage of different printer (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. Also the knowledge about "who the user is" is important (there might be different restrictions on printers' usage for teachers and for students or limit over the number of printed pages).

The second step is to choose the resource (printer) that best suits user's needs, taking in account the context information. Therefore, the location info and characteristics of all printers must be known to the party that takes the decision. As mentioned before the choice could be made locally or on the server.

If we consider a local case then all the information needed for the decision-making should be stored on the device. In a limited mobility the data might not be massive and might not endanger the availability of the device's 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. In this way though the system might omit some real time info, like the printers' queues at the current moment, thus producing poor solution (it is probably better to walk a few more steps to an empty printer, that quickly reaching a busy one).

Another issue has to be considered. To perform printing from a device to a certain printer the device needs a driver for that printer. A desktop computer can have installed on it drivers for all supported by the system/network printers, which for a mobile device is not suitable solution. 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.

A second possibility is 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. 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. 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.

There are two possible ways to ask the server to print a document: one is to pass to the server the current version of the document, and the information about the application that is using the document, and the other is to pass to the server a printable device-independent version of the document (such as a postscript file). The first solution requires the server to carry all possible programs and to recognize all possible file formats, which makes this option inconvenient. To achieve the second is much easier – one needs a postscript printer driver on the mobile device side, which produces a Postscript file, and then sends this file to the server. On the server side the file is printed on the chosen printer. 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.

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.

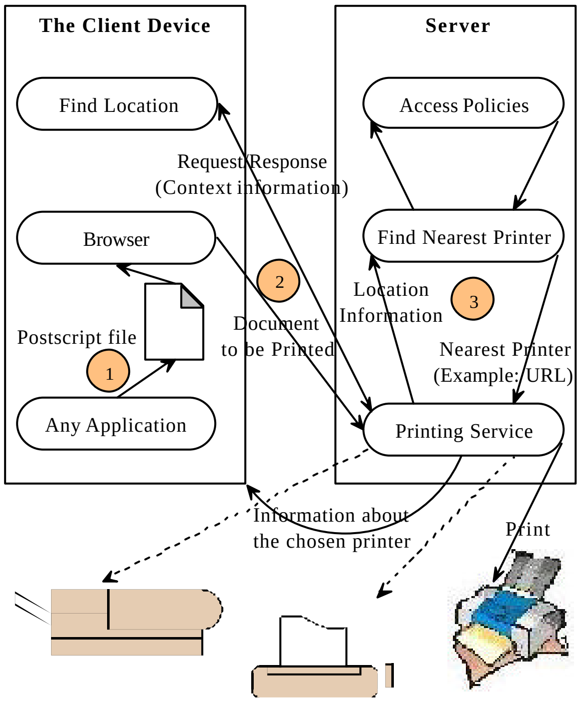

Figure 1: The printing process

We implemented a little less convenient but more immediate solution in which we perform the following steps: the user generates a postscript file (as we described it earlier), then he/she contacts explicitly through HTTP to the server, using a web-browser. The server provides a form, where the user points to the file and sends it to the server. An active component (e.g. a servlet) than opens a socket to the client and collects the context information. On this basis the server finds the nearest printer, prints the document and informs the user of the choice, together with explanation where the printer is located. The information is returned again in the browser via HTML page. 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. Figure 1 describes the whole process.

3. Positioning systems

To implement a location-aware system we need a proper positioning system. Different technologies are developed for determining the user's location. Lots of research had been done and systems had been made for automatically locating people, equipment, etc (Want, Hopper, Falcao and Gibbson 1992; WIPS 2000; Ward, Jones and Hopper 1997; Priyantha, Chakraborty and Balakrishnan 2000; Hightower, Boriello and Want 2000; Hightower and Boriello 2001). These different systems address different problems and so the location-sensing in each of them has different parameters, properties and accuracy. Some of them are suitable only for finding the position of the device when outdoors (Global Positioning System GPS), while other only work indoors. Additional infrastructure and/or equipment is necessary for most of the location determining systems (i.e. Active Badge and Active Bat systems require special tags and basis/stations; in the GPS case the infrastructure is already in place, so that it can be given for granted, but the user is required to have additional hardware on the client machine - a GPS receiver).

In general more appropriate would be a system that does not require additional hardware or infrastructure. In our system we use the IEEE 811b network that is already in place so it requires only adding a software layer. A small module on the mobile device connects sequentially to three or more access points in the wireless local network and measures the signal strength (the wireless network card acts as a sensor). Note that in a conceivably wireless networked city, such method would work indoors and outdoors. The results of the measurements are used to determine the position. There are different ways of doing this. With one of the methods called regression the position could be returned in physical coordinates (x, y, [z]) and in another (classification) as a more semantically meaningful (symbolic) expression, like floor and room number or “Professor X’s office”. Depends on the method chosen the accuracy varies. The research shows (Battiti, Brunato and Villani 2002) that the average error percentage in this second method is lower, thus it is more reliable.

The position-determining system that we use comes in two variants: the first returns raw data (physical coordinates); the second returns a semantic description of the location. The second option we find more suitable for our goal because we can explicitly take account of the local topology (meaning walls, aisles etc.).

In a system, where more than one positioning systems will be used one can think of introducing a semantic server, which translates data from the format used by the device (GPS, WLAN, Bluetooth) into format, proper for the server that offers the printing service.

4. Generalization

As we mentioned above the problem we address, i.e. the context-aware printing, can be viewed as a more broad-spectrum problem of providing context-aware services in common, as “printing” is a service and it uses other services – the APIs provided by the OS. For this, one needs to add a software layer that can be either on the mobile device or partially separated between the device and a server. The idea of having everything on the mobile side has some disadvantages: firstly the precious device resources, as memory, are endangered – the needed context information can be quite large, depending on the concrete application, and even sometimes it is impossible for the mobile component to be aware of all possible settings that are available in different places; and secondly (and probably more important) this might lead to missing some dynamic factors, thus providing weak solutions. However at least part of the context data is taken from the client machine so a server should contact it and download this data. After this point we have again two possible paths – to have the server provide the whole customized service, or to provide a “meta-service” – just point the solution and leave the rest to the client. We mentioned that providing a “meta-service” in our case is not appropriate, but in certain other cases it might be applicable and even suitable. Important issues are the performance, scalability and reliability of the system but one can fight them in standard ways. The most convenient for the user implementation would be the seamless one, but here we enter a less trivial and in lots of cases proprietary area (i.e. to integrate the system in a Learning Management System or the OS). One can overcome this situation in a little less convenient way of resolving the problem - two step process can reconduct (through the notion of a stub) to a conventional, local use 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).

5. Conclusion

In this paper we discuss context-aware services as supporting systems in the transition from e-learning to m-learning. We have implemented a working prototype to a location-dependent printing problem: how to print a document from a mobile device when being in a generic location. We selected the printing problem because of its practical value, and also because it is a prototype for a class of other problems that can be encountered when moving from a traditional environment to a mobile setting. We identified few different architectural approaches to achieving the same goal and comment their pros and cons. We believe that lots of the mobile devices' capabilities problems (i.e. lack of memory) could be overcome with a proper choice and that our architecture satisfies the requirements.

6. Acknowledgement

The work is partly supported by the WILMA project of the Provincia Autonoma di Trento.

7. References

Battiti R., Brunato M., Villani A., Statistical Learning Theory for Location Fingerprinting in Wireless LANs: *Technical report DIT-02-0086* University of Trento, October 2002

Hightower J., Boriello G.: Location Systems for Ubiquitous Computing, *IEEE Computer*, Aug. 2001, 57-66, available on line at http://www.intel-research.net/Publications/Seattle/062120021154_45.pdf

Hightower J., Boriello G., Want R., SpotON: An Indoor 3D Location Sensing Technology based on RF Signal Strength: *Technical Report #2000-02-02*, University of Washington, Feb. 2000

Priyantha N. B., Chakraborty A.,. Balakrishnan H, The Cricket Location-Support System*, Proc. 6. Annual Internat. Conf. on Mobile Computing and Networking*, Aug. 6.-11., 2000 Boston

SLL (2001), Mobile Learning Explorations at the Stanford Learning Lab: *A newsletter for Stanford academic community, Speaking of Computers* Issue 55, January 8, 2001, available on line at http://acomp.stanford.edu/acpubs/SOC/Back_Issues/SOC55/#3

Steinberger C., Wireless meets Wireline eLearning, *Proc. of ED-MEDIA 2002*, June 2002, Denver (CO,USA)

Want R., Hopper A., Falcao V., Gibbson J., The Active Badge Location System, *ACM Transactions on Information Systems*, Vol. 10, Jan. 1992, 91-102

Ward A., Jones A., Hopper A., A New Location Technique for the Active Office, *IEEE Personal Communications*, Vol. 4, Oct. 1997, 42-47

WIPS (2000), *WIPS Technical Documentation*, Royal Institute of Technology, Sweden, available on line at http://2g1319.ssvl.kth.se/2000/group12/technical.html

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