

## Technical working paper 2004, NKI Distance Education

### Exploring online services in an mobile environment

Written by Aleksander Dye, Truls Fagerberg and Bente Midtsveen.  
October 2004

This paper is written as a part of the project "Mobile Learning: The Next Generation of Learning" as a result of the research done at NKI Distance Education (NKI) for adapting SESAM (the learning management system (LMS) of NKI) to the mobile environment. The aim of this project is to develop and adapt courses for current and future mobile handsets where mobile learning is defined as the provision of training courses via wireless devices – Personal Digital Assistants (PDAs), smart phones and mobile telephones. These courses will be developed as part of a fully functional mobile Learning Management System (mLMS) and the courses will be tested and evaluated, and the results widely disseminated. The web page of the Leonardo project is found at:  
<http://learning.ericsson.net/mlearning2/index.shtml>.

This paper accounts for and discusses the challenges of mobilizing content from traditional clients to mobile users who use a PDA or similar devices. Smart phones and mobile telephones are not discussed in this article as the screen is much smaller and has other limitations that need special attention. When the term “mobile device” is used it refers to a handheld device that is capable of displaying hypertext mark-up language (html). This paper concerns PDAs used as a mobile client to an LMS, in our case, SESAM.

### *Summary*

This paper is a technical working paper written by developers at NKI Distance Education in the EU Leonardo project "Mobile Learning: The Next Generation of Learning", and describes the work carried out in the first year of the project. We have enhanced our existing LMS to serve mobile clients that has an html compatible web browser built-in. We have simulated the future, as we believe it might be, with a wireless always-online environment with five mobile users. The services we found interesting and useful were IP Telephony and messaging services such as MSN Messenger. We also reviewed different web browsers, but we ended up with Pocket Internet Explorer that comes with PDAs with the Windows Mobile operating system. We are looking forward to the development of Opera's Small-Screen Rendering™ browser for the PDA that hopefully will enhance the readability and screen rendering on small screens. Some of the most important findings were that if the structure of the document is good the challenge of transforming to small screens or different layouts was actually requiring less time and resources than expected. We did experience difficulties with large illustrations that contained too much information for a small screens as well as large fixed width flash animations.

### *The need for adapting to mobile devices.*

To better serve the increasing demand for different formats as well as mobility we need to give the mobile user a better experience with existing web pages. The reason for targeting mobile devices at this point is because there are more and more users who acquire mobile devices and wants to use them with our services. Today the market is small, but growing for mobile users – this is why we needed to adapt the existing web pages and services to the mobile devices. One of the biggest challenges concerning the mobile devices is find acceptable solutions adapted to the small screen size. There is simply not enough room for all the information found on a traditional web page on the small screen. Another problem is the limited data transfer rate and processing power found in mobile devices. When people use a mobile device with Internet connectivity the data rate is usually

much lower than one may expect from traditional clients. The mobile network restricts the data rate and if the user is accessing by a mobile network the cost involved is also a consideration. These challenges might be overcome with time, but until then, we must serve our mobile clients the best services possible with the minimum of effort. We might want to serve the mobile users an alternative page to that which is served to our traditional clients. There are some mobile browsers that could reduce the need for developing specific pages for the mobile client, this is not a standard at the time of writing.

### ***Always-online test environment***

For this work package we have established a wireless, broadband community counting 5 users of mobile learning. At home and at work, these users are equipped with PDAs that have wireless access to ADSL (Asymmetric Digital Subscriber Line – a broadband technology). This environment allows the users to explore wireless applications we consider to be generally available in the future. We have developed and tested out several applications and software that through this “always online” environment utilizes:

- **Synchronous communications, chat**
- **Quicker response on e-mail**
- **High bandwidth gives fast downloading of course content and use of audio, video and advanced graphics.**
- **Use of Flash, Java etc due to high storage capacity in future PocketPC**
- **Access to the resources at the Internet at all time**
- **ADSL gives you control over cost**
- **Not dependent of synchronization with desktop pc**
- **Online assessments and assignments**
- **Opens for collaboration between mobile learners**

We are using a wireless 802.11b base station connected to a broadband Internet connection that provides the basic “always online” infrastructure. We have tried out the wireless technology at home and at work to see how it influenced the way we utilized the Internet as a source for information as well as the benefits of studying wireless. We used an iPAQ PDA with wireless connection to the base station (or to a cell phone when we were on the road) as a terminal. The LMS used was NKI's SESAM that serves all our students on a daily basis. We modified some code on the development server so that our web pages also would fit the small screen of the PDA (240\*320).

### ***Browsers***

We tested NetFront v3.0 for Pocket PC and ThunderHawk, we also used Opera's Small-Screen Rendering™ technology in addition to the Pocket Internet Explorer (Pocket IE). After testing the different browsers and considering their pros and cons, we ended up with the original browser Pocket IE. This was the decision after arguing that our pages have to work on the most commonly used browser on the PDA. The other two browsers were lacking functionality and they were not free ware browsers. There is work in progress to improve these browsers and even though they had features that the Pocket IE did not have, for instance the option to alter between landscape/portrait modes, we decided to go with the market leading browser Pocket IE. To utilize flash technology we had to download a plug-in for the Pocket IE and it worked “out of the box”. In time we believe it will be installed by default. NetFront did support flash as standard but we could not find a plug-in that worked with ThunderHawk. We are also testing Opera and the Small-Screen Rendering™ technology they utilize on mobile phones and is available in the traditional browser by pressing <shift> + <F11>, this gives a view of the page with limited screen real-estate.

### ***Synchronous communications***

To explore the possibilities with synchronous communication we have tested two very different and exciting applications. One of the most interesting and useful of these might be the use of instant messages from the PDA by using MSN Messenger (MSN). MSN is Microsoft's popular instant messaging client that allows users to communicate with each other in real-time. This client lets you keep your existing contacts wherever you are, and if you add new contacts they are saved to a server that is maintained by Microsoft. The benefit is that wherever you log on, being from a desktop computer using the traditional MSN Messenger client, the web client or via the PDA Pocket Messenger, you have your contacts with you at all time. This is a great way to keep in touch with fellow students without being logged in to the LMS. This is an application many students already utilize and therefore it is possible it will be the preferred way of communication (second to e-mail) between students who wishes to cooperate. Recently numbers provided by the Norwegian MSN portal, states that there are approximately 900 000 Norwegian user of MSN Messenger.

We have also tested another form of communication, using the software Skype, which is an IP telephony application that is available for Windows, Mac OS X and Linux. There is also a version that is named Pocket Skype that works with PDAs with the Windows Mobile operating system. This allows the users to make phone calls over the Internet to anyone with a Skype account and has the same cost as being online in any other way. It is recommendable with a broadband Internet connection to avoid to much delay between the parties conduction a conversation. This has been tested between two students as well as a group conversation between three people and works surprisingly well. This might be a challenge to the long distance calls of today. Compared to MSN Messenger, Skype is so far in the very beginning and is not so widely known. The benefit tough of talking compared to typing is quite obvious. The possibility to have a conversation between fellow students even across countries is appealing to distance education, especially since the cost of the "always online" Internet connection is not an issue. This also benefits those students that are slow writers in a synchronous environment such as MSN. Another appealing development is the possibility to make calls to traditional phones for a low price via IP telephony.

### ***Using flash on the PDA***

One of the most interesting parts of this technical development has been the work with flash on the PDA. Flash is an authoring software developed by Macromedia, and is used for producing vector graphics-based animations. Flash makes it possible to develop various programs/features such as navigation interfaces, graphics illustrations, interactivity in a resizable file format that is small enough to stream across a normal modem connection. Flash is very suitable for use on the web due to the vector graphics ability to adapt and adjust to different display sizes and screen resolutions and it is designed for optimized delivery.

In this project we wanted to use flash for producing assignments mainly in the form of multiple choice and drag-and-drop. We have also developed parts of the courseware with the use of flash together with audio. This has been in a sort of pre-assignment or case as a basis for the real assignment that follows.

After experimenting with the course assignments developed in Flash, we adapted them to the PDA. The assignments have been adapted and further developed from the original version intended for use on a large screen to be used on the PDA with the Internet Explorer and the Flashplayer 6 plugin for PDA installed.

### Screen format on the PDA

We found that 240 \* 270 (width \* height) pixels was the optimal size and filled the screen as much as possible (see picture 1).

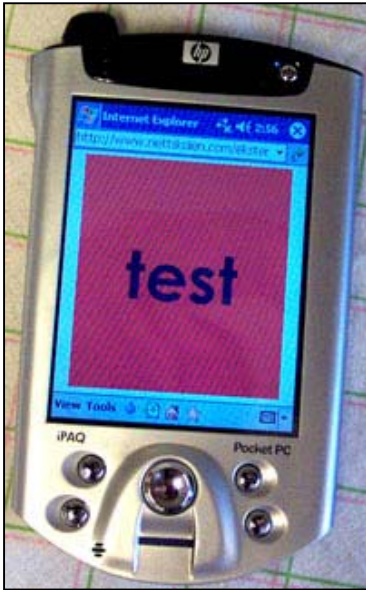


Figure 1: Full size (the red colour)

### Using text in Flash

When starting our developments we found that the text we used in the assignments was too small and nearly impossible to read when presented on the PDA. We started to experiment with different font types and sizes. By default Flash uses anti-aliasing on the font and that makes the text a little blurring in the edges. To get the text more sharpen and easy to read on a small screen, we found that we could use an option in flash call “dynamic text”. This proved better concerning readability and solved our problem.



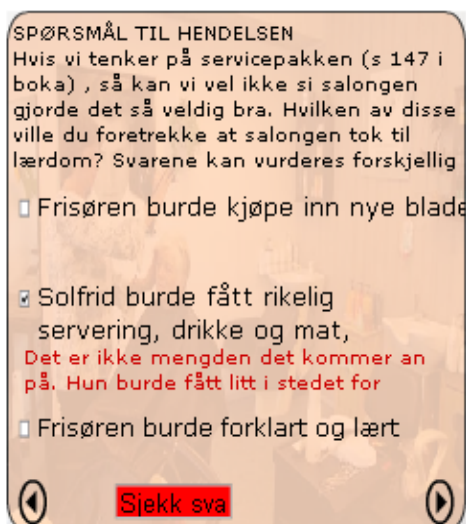
Figur 2: Different methods with text, verdana 12 pix.

On figure 2, the font is 12 pixels, but the flash program shows different sizes using two different methods. We have no good answer why Flash do so, but that's no special behaviour to PDA since this is how the program itself react. Later on, we also discovered we could keep the font static and use the options “Use Device Fonts” in flash. Flash uses device fonts to display certain text blocks, so that Flash does not embed the font for that text. This gave us the same output results as setting the text to “Dynamic” but we had more control over line breaking and text behaviour and it might also increase the file size.

We experienced a lot of difference in behaviour with the same flash file when using it on a PC versus the PDA. For instance we had textboxes that showed three lines on a PDA, could only shows

## Mobile learning: the next generation of learning NKI Distance Education

two lines on a PC. We solved this by adjusting the size of the textboxes to fit the PDA. We have also made a couple of assignments that use a “drag and drop” functionality. This worked very well on the PDA, but we experienced the same issues here regarding text readability. One must also have in mind that some functions, like “onMouseOver”, wouldn’t work on the PDA since the PDA is equipped with a touch screen and a stylus and no mouse like on the pc. Here are two screens shoots from these assignments made in Flash (in Norwegian).



Figur 3: Multiple Choice assignment



Figur 4: Drag and Drop assignment

### Video on the PDA

We have also tested out the use of video on the PDA with small video clips showing a discussion between two persons in a learning situation. This has worked very well using the Windows Media Player and we had no problems with viewing the files with high quality both on picture and audio. This is of course also due to high capacity of the wireless broadband solution.



Figur 5: Screen shot from a video clip

### **Synthetic speech on the PDA**

As a result of previous projects working with Universal Accessibility, we have also tested out the use of synthetic speech on the PDA. So far we are very pleased with the early results and the possibilities this technology gives us. We will describe the result and experiences from this work in the first working paper in the second year of the project.

### ***Identify target devices***

To serve the correct pages to different devices we need to know which devices we want to adjust our content to. The most basic question is; how many different kinds of devices do we want to give special attention? If the answer to this is none, everyone should get the same page, we have no need to identify the devices and everyone sees the same page. If we want to serve two different pages to different clients we could check for one property e.g. screen resolution (on the iPAQ: 240\*320); if the width of the available page of the client is less than a specified size, then serve page a, else serve page b. This could be accomplished by java script, but that is limited to the clients supporting java script, which are not all, and some might turn java scrip functionality off. If there are reasons for giving the users the choice between the different versions, e.g. bandwidth limitations could be a reason for wanting to access a light version of the page but still gives the basic functionality. If you would like to target more than two devices and/or give the choice to the server on which page to serve we need to know how to identify the wanted devices. This could be accomplished in different ways, but to ensure that we do it regardless of the device we want to have the server identify the client.

Identifying the client may be done by reading the http-header and serve a page based on the information retrieved. The http-header contains information about the user-agent among other information, but this is one of the possible variables we could use to distinguish between the pages we want to serve or base our layout on. If we do the testing on the server side, we know it will be adaptable to all types of clients, since the change is done on our server, and we know that if a client is to read pages from our server it needs to send a request and our server must send a response. This response could be based on the request and designed to fit the user-agent making the request for a page. We are considering a browser-sniffer called BrowserHawk that is an application that is installed on the web server to get information about the client accessing the web page. This could help us identifying the mobile devices as well as helping with either redirection to another web page or changing the layout of the page to better fit the small screen.

### ***Server or client based adaptation?***

This is in our opinion one of the most basic and fundamental choices to make. In our case we could not base our layout on the clients and their ability to handle the layout. If one bases everything on the client, there is no need to change the code if, for instance, the mobile browser is satisfactory for use on traditional web pages. This is an option if it is not business critical to serve the mobile client. If it is critical for the service, however, it has to be done on the server side. This is because this is where the business has control. We do not know what type of clients our users have and cannot tell the users to install programs to access our pages. This is possible only if the mobile services are supplied in addition to the traditional services, and if the users are aware of this fact. Then they will probably find the service as a nice addition to the other services the company has to offer. However, if the company is to gain new customers from supplying services for this medium, there is a need for a more professional approach.

By identifying the client on the server-side we have some different possibilities; we can serve a

different page than what we would do if a traditional client contacts the server or we can change the layout. We can serve a different css (style-sheet) or strip away tables to avoid sideways scrolling. By using the server as the identifying part we can assure that it will work and do not have difficulties concerning users who have disabled their java script, and so on.

### ***Concluding remarks***

We have tested some of the available browsers on the market today and are closely watching the developments taking place, specifically the development of the Opera browser for the PDA. There are quite some challenges in converting existing materials from the large screen of the desktop to the small screen of the PDA or other mobile device. The optimal solution for the small screen is obviously to make the most of the space available, and the challenge is to do this globally on the server so that one does not have to do this work on every single web page. If the structure is basically good, there is already technology on the market, which solves the problem with small screens quite well.

There are still a number of challenges concerning the use of Flash and large images because of the reduced readability when shrinking the image. This has not yet been solved, but there might be possible to zoom in and out of an image to see the details. The work we have done with Flash, video, synthetic speech and different communication forms so far, demonstrates that there are great potentials built into in these small pocket computers that just waits to be released. Combined with a wireless, always online environment these online services, we are strong believers that this could benefit both the online tutor and student in several ways. This will be further described and discussed in the second year of the project, were we will test out the didactic always-online environment for PDAs with real students.

## ***Resources***

Skype

<http://www.skype.com/>

BrowserHawk

<http://www.browserhawk.com/>

Opera

<http://www.opera.com>

MSN

<http://www.msn.no>

Media Queries:

<http://www.w3.org/TR/css3-mediaqueries/>

Netfront:

<http://www.netfront.no/>

ThunderHawk:

<http://www.bitstream.com/wireless/?wireless-web>

The World Wide Web Consortium (W3C):

<http://www.w3.org/>