Towards Pervasive Learning: WeLearn.Mobile

A CPS Package Viewer for Handhelds

Susanne Loidl

FIM – Institute for Information Processing and Microprocessor Technology

Johannes Kepler University Linz, Altenbergerstr. 69, A-4040 Linz, Austria

loidl@fim.uni-linz.ac.at

Abstract

Smart applications are influencing our life more and more. This paper presents an application where pervasive computing is used in the context of e-Learning. First a short introduction to the ongoing changes concerning pervasive computing and particularly e-Learning is given, followed by a report on an e-Learning framework used in schools, universities and the area of continuing adult education. This framework is the basis for the viewer application, which provides the possibility to present CPS (Content Packaging Specification) packages on handhelds (namely Pocket PCs). The paper concludes with an account of possible extensions and prospective future work.

Keywords: E-Learning, Pervasive Computing, Pervasive Learning

1 Introduction

Pervasive Computing penetrates our daily life. We encounter “smart” applications in various areas, such as our workplace (e.g. personnel tracking, intelligent facility management, etc.), our home (e.g. appliance control), our leisure time (e.g. recording of road taxes, etc.), and the range of applications will certainly increase over the next few years through the adoption of intelligent agent technologies.

Moreover, when we talk about pervasive computing, we usually think of anywhere, anytime, any data and any device. This means:

- Anywhere: Global accessibility, especially with regard to various kinds of communication networks

- Anytime: twenty-four-seven, but also independent of other services or persons (asynchronicity)
- Any data: email, public services, inter- and intranet, various data formats
- Any device: (Tablet-) PC, Personal Digital Assistant (PDA), cell phone, …

When we think of our working environment this vision has already come true. We are permanently connected to the internet either by cable or by using e.g. WLAN if on the move. We use our desktops, laptops, handhelds and even cell phones to search for, check and update documents as well as personal/business data or customer information. We just take these smart applications for granted.

On the other hand, this also means that our day-to-day life becomes faster and faster. Product life cycles decrease and to the same extent also the validity of knowledge falls. Consequently, life long learning is becoming more and more important. The days are gone when one could “survive” one’s whole life, or at least one’s whole working life with the knowledge one learned in school or at the university.

One major change is that schools and universities are now not only teaching students attending the university from eight to five, but should also provide for education of much older people who require information on small and specialized topics and who must fit this into their regular work life [11] (continuous and just-in-time learning, [14]). Also, these might not be accustomed to new kinds of learning, requiring assistance through human or electronic coaches.

Keeping the following in mind:

- New technologies, new media, new problems
- A knowledge based industry with rapidly changing knowledge
- The globalisation of markets and high-level jobs

- The need of soft skills, such as time management, communication skills, etc.,
- And especially the need for knowledge acquisition anytime and anywhere

It is therefore not surprising that schools and universities are more and more focusing on adding e-Learning to their teaching portfolio.

E-Learning is becoming a fixed part of most people's life, as they are forced to life-long learning. It is also a fast developing area. In the last few years hundreds of platforms ([1], [2]), tools, electronic learning material, and so on have been developed. In parallel, a lot of time has been spent on the development of appropriate specifications and standards ([3], [4]) to ensure interoperability especially for materials. The latter is especially important, as e-Learning material is much more costly to develop [12], therefore the need for reuse also increases.

The increased need for learning can also lead to problems of scale: Even though E-Learning was intended to be performed without teacher support, results for this proved to be suboptimal. Support by coaches or teachers is still required, although these are scarce at university level.

2 First steps towards pervasive learning

The Institute for Information Processing and Microprocessor Technology (FIM) has been working for a long time in the area of E-Learning. Within the last three years the WeLearn.Framework (Web environment for Learning; [5], [6]) was developed.

2.1 WeLearn.System

The first tool developed in this direction was WeLearn.System (WS). It is a free and open virtual learning environment. Within its implementation several aims were followed, keeping the four “anys" in mind:

- Accessibility: WS is implemented in pure Java and is web-based, so that it can run on most systems and is accessible from anywhere and at any time. Users can access their courses within WS from all over the world simply through a web browser.
- Easy to use: Well-known paradigms like an explorer-like style or an icon toolbar are employed to keep the handling as simple and intuitive as possible.
- As few limitations as possible on any axis: This means not only that WS is useable for any teaching style or model (it can be used with any didactical model, target group or topical area), but also that any data format can be imported. All kinds of document formats (text, audio, video, etc.) can be uploaded and accessed. Our experiences show that many lecturers already possess electronic teaching material of some type, ranging from static presentations to interactive models. They can easily include them into WS without a need for transformation and start teaching in the virtual teaching/learning environment immediately.
- Use of standards: In order to guarantee the longevity of WS and any teaching material developed for use within and therefore avoiding stranded investments, WS supports various E-Learning standards, such as the Content Packaging (CPS, [7], [8]) or the Metadata specification (MD, [15]) of IMS. CPS is a specification, which defines the structure of courses, couples hierarchical structure with physical resources, and assures portability and re-usability (for details see section 3.1).

The idea behind this virtual environment is, not only to present or study electronic learning material, but also to debate with other people, to learn about different viewpoints, to communicate and to interact with each other learners as well as with coaches and teachers.

2.2 Limitations of this approach

One problem of WS is that learners must be online for using any of these functions, even for the simple presentation of material. Consequently, an offline solution was investigated and implemented, the WeLearn.OfflineConverter. Through this, the aspects of "where" and "when" were improved by obviating the need for a constant online connection. This first approach to being Internet-independent supports burning courses onto CD-ROMs or DVDs.

Unfortunately, this only shifts the problem to the dependency on an available and appropriate disc drive – and therefore to a desktop or laptop computer. Both, even the latter, are not perfectly suitable for learning "on the move". They are rather for stationary learning, with more or less ease of changing the location. Therefore, support for learning through handheld devices is needed to improve portability in the sense of pervasive learning. As for them a permanent network connection cannot be expected, this problem must be coped with, at the same time resulting in independency from a permanent Internet connection, one of the other shortcomings noted above.

2.3 Solution: WeLearn.Mobile

This led us to a more “mobile” solution - a tool that allows an offline presentation of courses in CPS-format on a handheld, called the WeLearn viewer for CPS packages on handhelds, or shorter WeLearn.Mobile (WM). For this a connection can either be available, widening only the choice of devices supported, or not, providing in addition connection independence. The result is the possibility to use learning material designed for online courses on full and stationary PCs also on small and portable devices (see also [13]).

Handhelds are important as target devices in the context of pervasive computing because they are becoming more and more widespread. They are small, easy to use and can easily be connected to desktop PCs. Moreover, as IBM states [9]: “The proliferation of wireless devices such as PDAs and smartphones has made pervasive computing a reality for consumers. The use of wireless devices is the beginning of a new way to access information. The ever-increasing mobility and accessibility afforded through pervasive computing helps customers and enterprises to: reduce costs, integrate processes, create new revenue sources, grow their businesses. Pervasive computing helps to bring about changes in lifestyle as new applications and services become available to consumers.”

They are additionally becoming increasingly important in the context of learning as several studies showed the benefits of handhelds in the context of learning. In short, the key results of e.g. the Palm Education Pioneer Evaluation [10] states that "the PEP evaluation study found that handheld computers can offer unique benefits to students and teachers. Students can have a personal, portable device ready-at-hand for individual or collaborative learning activities, wherever they go." and results in the following: "Key findings include: 89% said they found the handhelds to be an effective instructional tool for teachers. 93% said they believe that handhelds can have a positive impact on students' learning. 95% said their students were "very" (66%) or "fairly" (29%) comfortable using the handhelds. 90% plan to continue using handhelds in the classroom post-study. 72% said handhelds are more easily used in the 'flow of classroom activity' than desktop computers.”

3 WeLearn.Mobile in detail

One important issue with handhelds is adherence to standards. Because of their inherent limitations (e.g. small screen size, limited "keyboard"), applications and file formats tend to be different from "full" computers. However, this is obviously not a good idea in terms of compatibility and data transfer. A main aspect is therefore support for the "big" applications also on these "small" devices, like the pocket versions of Microsoft Word or Excel.

Even better is, however, supporting open standards so not only specific applications interoperate, but also different ones: Companies and product developers want to increase their return on investment, as well as to eliminate redundant developments and to deliver learning material across multiple media (online, CD-ROM, print, etc., but also handhelds). “As organizations make significant investments in digital learning content, there is a strong desire to have greater assurances, portability, and re-usability. As organizations focus on providing learners with the 'just right' content and activities, there is a strong desire to have the ability to more easily store, search, index, deploy, assemble and revise content. All of these hopes are part of the story of 'learning standards' [4]”. Especially, re-usability and portability are important issues and for example the CPS of IMS is a de-facto standard in the context of e-Learning.

In the following, this standard is presented briefly and afterwards the implementation of WM is described in detail.

3.1 Content Packaging Specification (CPS)

Before going into implementation details of WeLearn.Mobile, a short introduction to the Content Packaging Specification [7] of IMS is given, to provide an idea of how the learning material is organized.

The Content Packaging Specification consists of two major components: A so-called “Manifest File” and the resources (the physical files with the actual learning content).

The Manifest File (called "imsmanifest.xml") describes the (hierarchical) course organization in XML. Each manifest consists of a metadata section (information about the course, e.g. author, description, language, keywords, etc.), an organizations section (the structure of the course), a resource section (references to the physical files) and optional (sub)manifests. The CPS separates physical learning resources from their actual structure or presentation, allowing more than one use within different contexts or uses, i.e. in several courses.

The following snippet shows how this is encoded in XML. First each item is associated with a unique identifier (e.g. "MANIFEST01_ITEM22"), several attributes (e.g. isvisible), the identifier of the resource linked to the item (e.g. "MANIFEST01_RESOURCE102"), and the items title (e.g. "Introduction"). At the end of the listing of the items the resources themselves are specified by their unique identifiers (e.g. "MANIFEST01_RESOURCE102"), the type (e.g. "webcontent"), their file name, and where they are located (e.g. "chapter_1\ppt\intro.ppt"):

```
.........
<item identifier="MANIFEST01_ITEM22" isvisible="1"
      parameters="" identifierref="MANIFEST01_RESOURCE102">
   <title>Introduction</title>
</item>
<item identifier="MANIFEST01_ITEM23" isvisible="1"
      parameters="" identifierref="MANIFEST01_RESOURCE103">
   <title>Algorithms</title>
</item>
.........
<resource identifier="MANIFEST01_RESOURCE102"
       type="webcontent" href="chapter_1\ppt\intro.ppt">
          <file href="chapter_1\ppt\intro.ppt" />
</resource>
<resource identifier="MANIFEST01_RESOURCE103"
       type="webcontent" href="chapter_1\pdf\algorithms.pdf">
          <file href="chapter_1\pdf\algorithms.pdf" />
</resource>
.........
```

In the design of such a CPS viewer for handhelds several things have to be considered. One reason why handhelds are widespread is that they are small, light-weight (much more than a desktop/laptop or tablet PC could ever be) and therefore easy to take along. Consequently, in contrast to them, handhelds possess a substantially smaller screen, a “weak” processor, less storage space and a limited battery capacity. All these restrictions must be considered in realizing a universally applicable presentation of e-Learning courses.

A limitation in screen size means that the user interface must be considered very carefully. Of course, in each program the user interface plays a major part, but when we talk about handhelds we are talking about a diagonal screen size of about 3,5 – 4,1 inch with a resolution ranging from 160 x 160 to a maximum of 320 x 480 pixels. Hence, we implemented several prototypes of the interface. These were evaluated by several test users. In the end:

- Input limited to necessary tasks: Text input is reduced to a minimum, e.g. available courses are listed automatically and the user can select them by clicking on the name, instead of typing the name, which can be awkward on handhelds.
- One click: Important information or frequently used features are always available on the first page. E.g. once a course is chosen, the course structure is always shown on top and the content of the files opens by clicking on the title of the (sub)chapter.
- Small and polished interface: Finally, the interface was optimised, so that handling is easy to learn and to remember, i.e. colours are used to give status information (file is available or not) instead of additional text or icons.

Another limitation is given by the storage space. When we talk about e-Learning courses and e-Learning material (text, audio, video, etc.), it is not unusual for a course to fill several hundred MB. Normally, this is no problem when talking about desktop devices or laptops, but in the case of handhelds this means that you have to find a solution for this, especially concerning CPS-packages, as this means that one does not have various documents, independent from each other and unstructured: As mentioned before, a CPS package represents a whole structured and homogenous course. In addition, the manifest snippet in section 3.1. shows that not only the course structure and corresponding files are described, but also additional information can be given, such as parameters for the presentation tool. Furthermore, it is possible that the resource section contains resources that are not (currently) referenced by any items. This is often the case when the development of a course is in progress and documents within the manifest are replaced by others. Then the original documents are not referenced anymore, but they are still listed in the resource section. Consequently, the data volume becomes larger than normally necessary.

One can not ignore this for various reasons. First, as said before the storage capacity of the handheld (although expandable through memory extension cards) may not be sufficient. Secondly, a bigger data volume will also result in more calculation time (remember: “weak” processor) and therefore increased response time. Moreover, of course battery capacity is wasted.

Hence, the presentation process is split into two separate phases:

- Transformation of the material: The CPS-package is transformed - unnecessary information is deleted and additionally required information, such as file size, is added (see section 3.3.1 for details). This is done on the desktop using the better processor and larger storage capacities. Only unavoidable calculations are performed on the handheld.
- Showing the structure: Only the structure and the size of the course are shown and the user can decide whether to download the whole course or individual parts/files.

In addition, a versioning function was implemented. This feature helps the user administering its learning material. If a newer version of the course (or of a file within the course) is offered, the user receives automatically appropriate information from the application and can then choose to keep the current version or replace it with the new one.

Content size is also of importance regarding the transfer to the handheld. When using a docking station, this is usually no difficulty as transfer speeds are high. However, when considering future extensions to Wireless Lan (WLAN) connectivity (see section 5.3) or other transfer methods, this becomes very a necessity to avoid undue delays.

3.3 WeLearn.Mobile implementation

Within WeLearn.Mobile three applications participate in the process of presenting courses (see Figure 1):

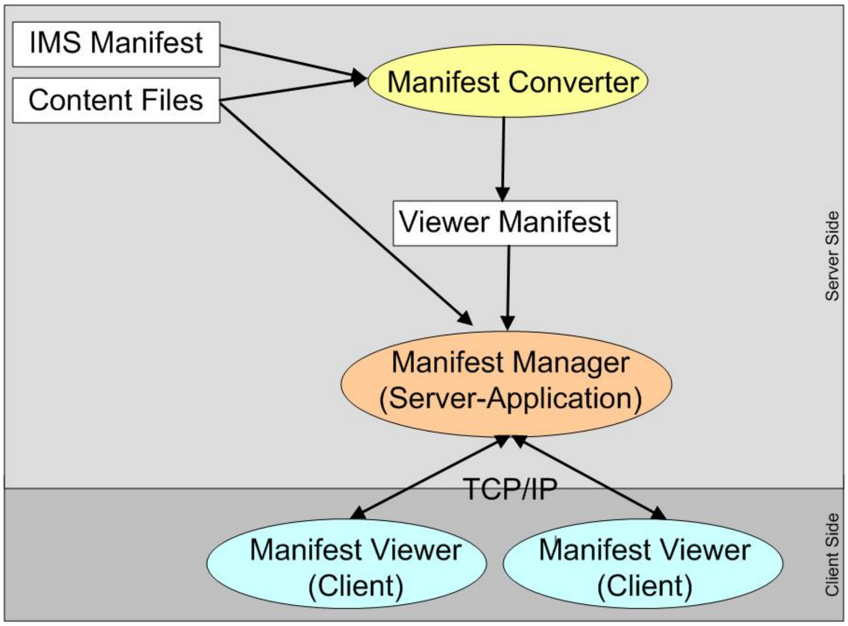


Figure 1: From a CPS package to presenting a course on a handheld

1. The original IMS manifest and the content files are parsed by the Manifest Converter on the desktop. The Converter produces a viewer manifest with additional information (size of the whole course, size of the files, creation date, etc.). One or more converted manifests can then be added to the Manifest Manager.
2. The Manifest Manager (also a desktop application) knows which courses are available and where these are located. When it receives a request from a Manifest Viewer (running on the handheld), the Manifest Manager sends the Manifest Viewer the manifests and the corresponding files in question.

3. Finally, the Manifest Viewer is responsible for presenting the course structure (described by the manifest) and the actual contents of the course material (files).

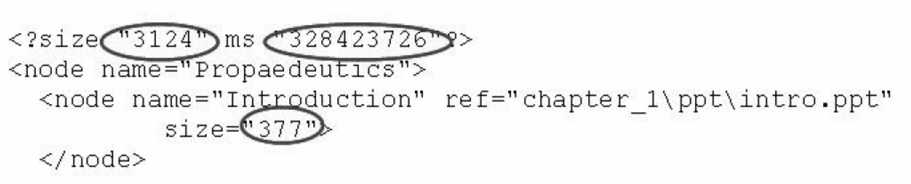
The next sections present a more detailed description of these three applications and their functionalities.

3.3.1 The Manifest Converter

The first step in the process is converting the IMS manifest and the content files to a handheld viewer conformant manifest.

Therefore the Manifest Converter was developed, which modifies the original IMS manifest. In particular the following actions are performed:

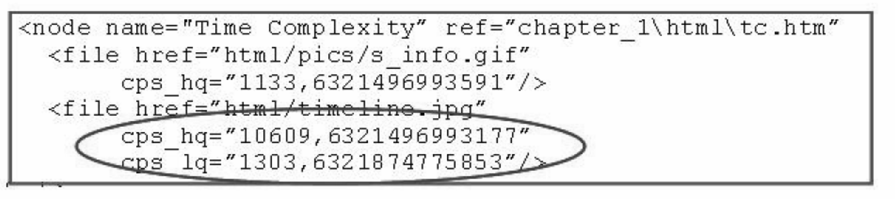
- Resources not referenced by items are deleted. This ensures that only files, which are definitely part of the course, must be transferred to and stored on the handheld.
- Irrelevant information, like some attributes, are deleted as well.
- A course title is derived from the manifest or generated if there is no title included.
- Attributes such as size of the package, size of each individual file and creation date (for versioning), are added (see Figure 2).



```
<?size "3124" ms "328423726"?>
<node name="Propaedeutics">
  <node name="Introduction" ref="chapter_1\ppt\intro.ppt"
          size="377">
  </node>
```

Figure 2: Manifest with modified attributes

- Additional tags are generated to ensure that the content can be shown properly: On the one hand this concerns HTML pages and embedded files (gifs, jpegs, etc.). Due to storage space limitations of small devices normally not the whole course, but pieces, i.e. single chapters, are stored on them. Therefore, all resources, not only the HTML page as such but also the files included within them, must be listed in the manifest (see Figure 3, text in the rectangle). Secondly, the compressing of JPEGs is possible. Again, in order to save memory, the user can choose to compress JPEGs. Then information about the original (high quality) file size (in attribute "cps_hq") and additionally also the compressed file size (in attribute "cps_lq") is stored in the manifest (see Figure 3, text in ellipsis).



```
<node name="Time Complexity” ref=”chapter_1\html\tc.htm”
  <file href=”html/pics/s_info.gif”
       cps_hq=”1133,6321496993591”/>
  <file href=”html/timeline.jpg”
       cps_hq=”10609,6321496993177”
       cps_lq=”1303,6321874775853”/>
```

Figure 3: Additional file information

- The Manifest Converter also checks whether all necessary files are included in the package.

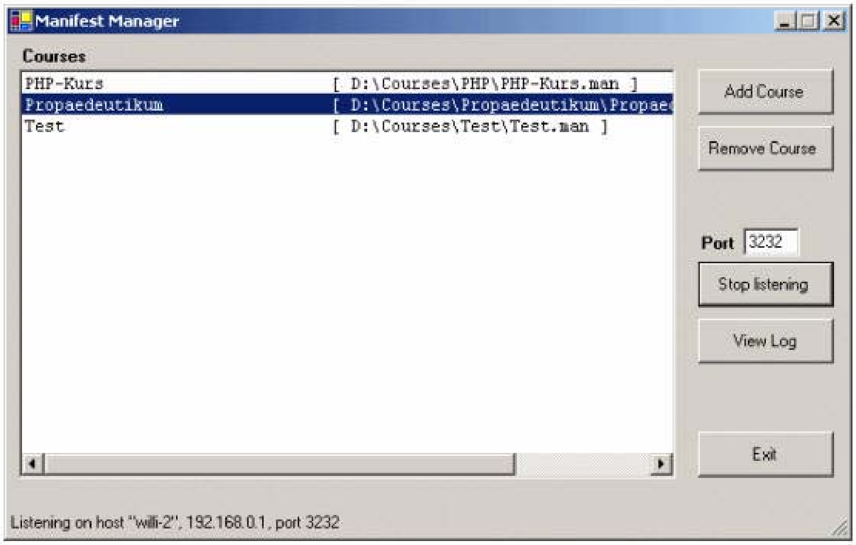
This results in the following manifest (in comparison to the snippet above):

```
<?size "3124" ms "328423726"?>
<node name="Propaedeutic">
  <node name="Introduction" ref="chapter_1\ppt\intro.ppt"
          size="377">
  </node>
  <node name="Algorithms" ref="chapter_1\pdf\algorithms.pdf"
          size="2127">
  </node>
  <node name="Time Complexity” ref=”chapter_1\html\tc.htm”
    <file href=”html/pics/s_info.gif”
         cps_hq=”1133,6321496993591”/>
    <file href=”html/timeline.jpg”
         cps_hq=”10609,6321496993177”
         cps_lq=”1303,6321874775853”/>
  </node>
.....
```

The reasons for these modifications are to provide the Handheld Viewer with all required information, but not more (as mentioned before "ordinary" CPS manifests can be quite large for complete courses). Consequently, this leads to an accelerated data calculation and transport to the handheld and also allows the user to download whole courses as well as course parts. Thus, the learner can concentrate on specific contents and store just the corresponding course material on the handheld, leaving out the rest and conserving storage through this. The same is true for removing files. Once documents are not of interest anymore, they can be deleted without having to delete the whole course.

3.3.2 The Manifest Manager

The Manifest Manager (Figure 4) represents the server side (desktop) of the process, connecting to the client (handheld) application (the Manifest Viewer).



```
PHP-Kurs                    [ D:\Courses\PHP\PHP-Kurs.man ]
Propaedeutikum              [ D:\Courses\Propaedeutikum\Propae
Test                        [ D:\Courses\Test\Test.man ]
```

Port 3232

Listening on host "willi-2", 192.168.0.1, port 3232

Figure 4: The Manifest Manager

The core tasks of the Manifest Manager (MM) are:

- Registration of courses (pointing to the available courses)

- Communication with handheld clients (file transfer via TCP/IP)
- Logging (of the communication events)

By adding a manifest to the Manifest Manager a course becomes available for download to handhelds. The Manifest Viewer (installed on the handheld) can connect to the MM and request either (a) all available courses, or (b) specific course material. In the first case, the MM sends a list of all available courses (all courses that were transformed into an appropriate manifest and registered with it).

In case (b), the user has already downloaded a course manifest to the handheld and now the MM sends only the files the user (the Manifest Viewer) asked for.

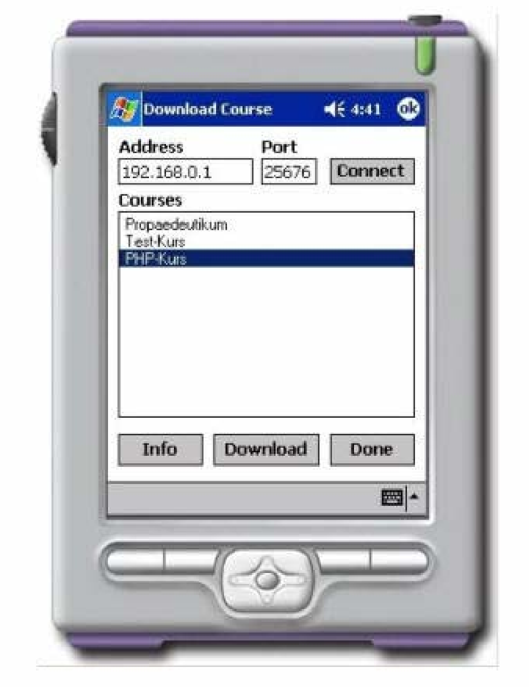
At the moment, no state of the client is maintained with the MM, as users can delete content on the handheld at any time and this information is not sent back in a kind of "synchronization". This would be required for introducing agent support for automatic content managing (see also at the end).

3.3.3 The Manifest Viewer

The last element in the presentation process is the Manifest Viewer (MV). It is the application the learner is using, whereas the Manifest Converter and the Manifest Manager are used by the course developers or providers, respectively. Currently, the connection between these components works via the normal sync-mechanism using the docking station. The docking station is a hardware connection between the server (running the Manifest Manager) and the handheld (running the Manifest Viewer).

The Manifest Viewers features are:

1. Showing available courses: By connecting to the Manifest Manager the MV retrieves a list of all available courses. The user can then select from one up to all of the courses (Figure 5, the Manifest Viewer running on a Pocket PC).



Address 192.168.0.1

Port 25676

Courses

Propaedeutikum

Test-Kurs

PHP-Kurs

Figure 5: Courses available and downloading a manifest

2. Loading and deleting course manifests: Once a course has been chosen, the manifest is downloaded to the handheld. Likewise existing manifests on the handheld can be deleted.
3. The presentation of the course structure: If a manifest is loaded, the Manifest Viewer parses it and creates an HTML presentation of the course structure (Figure 6), presented through a web browser.
4. Loading and deleting course material: The users can choose between downloading a whole course (then all course files are downloaded from the server to the handheld; possible only if there is enough storage available), a chapter (all files contained within the chapter are downloaded) or they can choose one or several unrelated files and download

only those interesting at the moment (or respectively according to how much free storage the handheld has).

5. If the user is no longer interested in some files (say a chapter) anymore, they can delete the corresponding files by selecting the individual files or whole chapters. Simply by clicking on the name of a chapter or the title of the course and selecting “Delete”, all files contained in the chapter/course are deleted.
6. The presentation of the course material: If a file is available (downloaded to the handheld) the corresponding manifest item in the structure is shown in colour (see Figure 6b). Otherwise the item is greyed (see Figure 6a). By clicking on the items name e.g. by selecting “3. Kapitel”, the HTML file containing the content is opened and presented to the user.

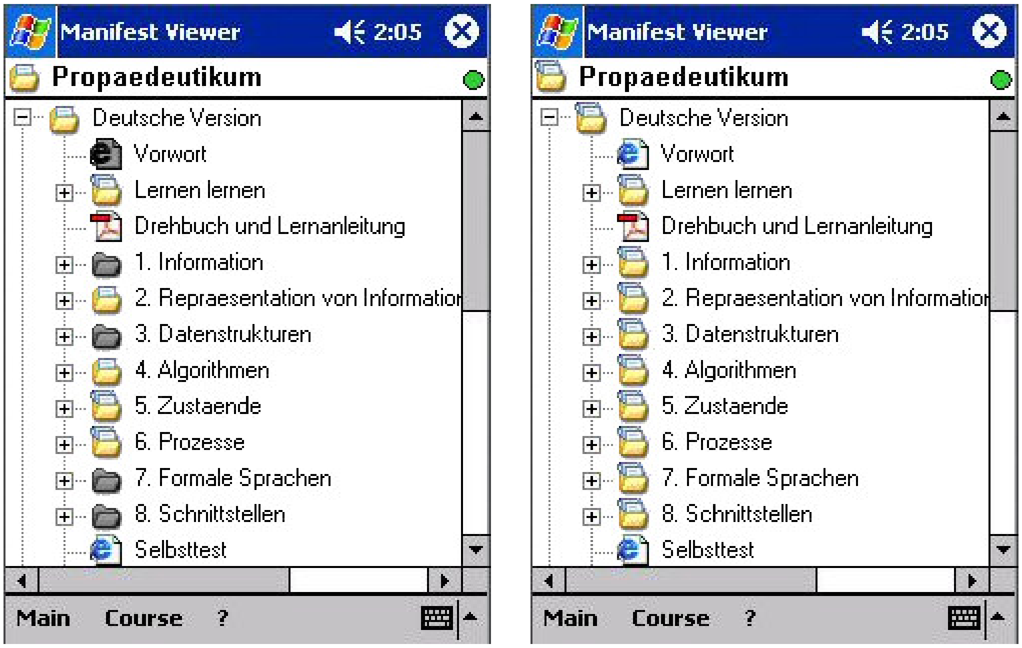


Figure 6: The presentation of the course structure: (a, left) a course where only some files are included, and (b, right) a course with all files downloaded to the handheld

7. The presentation of additional information (metadata, memory requirements, etc.): If metadata is included in the original IMS manifest, this information is also available in the Manifest Viewer. By clicking on the name of the course/item, the related metadata, or at least the size of the manifest/file, is shown. This information can assist the user in deciding which files to load or delete.

4 Practical evaluation

“Propaedeutic in Informatics” is a compulsory course for informatics students in their first semester at the Johannes Kepler University Linz. For several years, it has operated with blended learning as a didactic model. This means, that lectures and phases of self-organized study alternate. For the self-organized study normally the students have access to the WeLearn system (all material is included there and discussion fora are available), handouts with the printed texts and a CD-ROM is given to them. Additionally, in the winter semester 2004/05 the students also had the opportunity to work with WeLearn.Mobile. The Manifest Viewer was installed on the students´ Pocket PCs (several students already possessed a Pocket PC and to other interested students a Pocket PC was lent) and a short introduction in handling the viewer was given. Afterwards the usage of the viewer and the material (frequency of studying on the Pocket PC, of downloading the material, etc.) was up to the students, no further instruction or advice was given. However, they were obliged to write a protocol about their experiences with the viewer (in which way and how often they used it, how often they downloaded new material, how easy to use it was, proposals for improvement, etc.). The protocols/feedback were very positive and showed that all students enjoyed the flexibility of study possible e.g. while travelling, and, unexpectedly, the viewer was used several times a day. As possible improvements, mainly two aspects dominated: a) an increase in connectivity (the usage of WLAN connections) and b) a connection to the WeLearn system and a synchronisation of forum messages were mentioned several times (see next section).

5 Discussion of results

Although the system has been implemented completely and the practical evaluation resulted in a success, further work is possible and partially planned. In addition, some of the design issues discussed above and solved in WeLearn.Mobile are applicable to a wider range of applications.

5.1 Possibilities for agent support

One possible extension is the introduction of intelligent agents. These are already part of the stationary system, which is coupled to an agent platform to provide additional services. As already mentioned, one of the tasks could be helping users to decide which parts of the course to upload to the handheld (through server-side agents), or which to remove (from the client=handheld side). Guidance could be based on actions of other users respectively based on the progress through the already downloaded material as well as any cross-references made in the content which were available (material regularly visited should not be deleted) or not (and should therefore be downloaded).

Another area of the existing online platform are discussion fora. These are currently not part of WeLearn.Mobile. Supporting them would also be a field for future extensions. Offline composition of messages and posting them upon the next synchronisation is quite easy to realize, but selecting which messages to actually download to a handheld is much trickier. Especially so because one of the requirements described below does not really apply to them, as messages occur in large numbers, but are rather small individually. Also their numbers render manual selection for download practically unusable. Agents could provide support here through a pre-selection. Rule examples could be selection according to messages authored by the handheld owner, whose "ancestry" as well as any "replies" are probably important. Other ways of identifying user's interest could also improve the guidance for the user.

5.2 Transfer to other types of handhelds

The Manifest Converter and the Manifest Manager are implemented to be as platform independent as possible (usage on several operating systems is therefore no problem). Within the Manifest Viewer Pocket PCs were chosen as the target platform. More precisely, MV´s implementation is based on the Microsoft .NET Compact Framework. This is because the .NET Compact Framework is supported on Windows Mobile 2000-based Pocket PC, Windows Mobile 2002-based Pocket PC, Windows Mobile 2003-based Pocket PCs, Windows Mobile-based Smartphones and embedded systems running Windows CE .NET 4.1 and later. Consequently, a huge number of devices can be serviced with the Manifest Viewer at once and for other devices running Palm OS, Blackberry OS, etc. only one part of WeLearn.Mobile, namely the Manifest Viewer must be adopted.

5.3 New kinds of connectivity

One functionality discussed for future extensions is going in the direction of Bluetooth, so that the connection between the Manifest Manager and the Manifest Viewer can be performed using the cell phone functionality, which is already integrated in some handhelds.

Another option is to use WLANs. As a campus-wide WLAN is already in place at the JKU, it certainly makes sense to develop a slightly modified interface, which allows connecting to the Manifest Manager and downloading course material via this network. This will increase the ease of updating and the degrees of freedom in choosing where to learn.

5.4 Applicability to other domains

The concept of incremental presentation might also be useful in other domains. Direct transferability to all kinds of data presentation is possible. The following issues must be considered, however:

- Content must be partitionable: If the content cannot really be separated into mostly independent elements, incremental presentation is impossible. The main document must therefore be structured as a loosely connected collection of items. Hierarchical ordering will work best, while networks of elements will pose problems of "stale" (=not currently present on the handheld) links.
- Individual elements must take some time for "consumption". If granularity with regard to time is too small, re-loading and removal of elements takes too much time (and might not be available when needed), unless automation is supported e.g. through intelligent agents. Also, the number of elements will then increase, forcing the user to perform "micro-management".
- Required content must be predictable: Either, the system, through automated sub modules or the user must be able to predict to some degree the parts that will be required in the immediate future. This concept embraces asynchronicity, so access must be predetermined during online time to be available when needed offline. Therefore, sequential access improves this method’s utility, while random access prevents its usefulness.

No general requirement is separating a (probably often required) pre-processing stage from the actual content server. Although they must be conceptually separated, implementations might integrate them into one system. In this example both stages are separate, as multiple servers

(Manifest Managers) might exist, while conversion need only take place once (and might require significant resources, e.g. because of image resampling).

6 Conclusions

Pervasive computing becomes part of our everyday life more and more and the same is true for e-Learning. The WeLearn system and the WeLearn Viewer for CPS-Packages on Handhelds presented in this paper are the first steps combining these challenging areas.

Another aim for the future is reducing the number of the now three involved applications (the Manifest Converter, the Manifest Manager and the Manifest Viewer). At least it should be possible to just have one server and one client application to enhance user friendliness and usability for system managers.

Lastly, we are thinking of integrating this feature into WeLearn.System. Learners will mainly use the platform for learning, because within the platform not only the learning material is presented, but also communication and interaction takes place. Learning groups and individual learners can debate and work within the platform. In the future, it should be also possible to choose online within the courses, which courses (course files) they want to take with them on the handheld while they are working online in the platform. Consequently, the two worlds (platform and additional devices), will become one world for the user, and another step is taken towards pervasive learning.

Acknowledgment

This paper is a result of the project "Integrating Agents into Teleteaching-Webportals" sponsored by the FWF of Austria (Fund for the support of scientific research; Project number P15947-N04).

References

1. Baumgartner P, Häfele H, Maier-Häfele K. Auswahl von Lernplattformen, Studienverlag, Innsbruck, 2002, ISBN 3-7067.1771-X

2. Schulmeister R. Lernplattformen für das virtuelle Lernen, Oldenburg Wissenschafts-verlag, München, 2003, ISBN 3-486-27250-0
3. Loidl S, Paramythis A. Distance Education – A Battlefield for Standards. In: A. Szücs, E. Wagner, C. Tsolakidis (Eds.): The Quality Dialogue. Integrating Quality Cultures in Flexible, Distance and e-Learning. Proceedings of the 2003 EDEN Annual Conference, Rhodes, p. 89-94, 2003
4. The MASIE Center e-Learning Consortium. Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption, 2003, http://www.masie.com/standards/S3_Guide.pdf
5. Paramythis A. Project Report: Emerald (WeLearn 3.0) 2004, for former versions see also: Divotkey R, Mühlbacher J.R, Loidl S, Remplbauer D. The WeLearn Distance Teaching Framework. EDEN (European Distance Education Network), 2002 Eden Annual Conference, Open and Distance Learning in Europe and Beyond Rethinking International Co-operation, Conference Proceedings, p. 308-314, 2002
6. Sonntag M, Loidl S. Cooperative Agent-Supported Learning with WeLearn. In: G. Chroust, C. Hofer (Eds.): Euromicro 2003. New Waves in System Architecture. Proceedings of the 29th Euromicro Conference. Los Alamitos, IEEE Computer Society 2003, p. 157-164, ISBN 0-7695-1996-2
7. IMS Global Learning Consortium. Content Packaging Specification Version 1.1.3 - Final Specification, 2004, http://www.imsglobal.org/content/packaging/index.cfm
8. Mühlbacher J.R, Sonntag M. Roadmaps - Navigational Aids and Tools for Reusing Content in Distance Education. In: A. Szücs, E. Wagner, C. Tsolakidis (Eds.): The Quality Dialogue. Integrating Quality Cultures in Flexible, Distance and e-Learning. Proceedings of the 2003 EDEN Annual Conference, Rhodes, p. 73-78, 2003

9. IBM. Pervasive computing - From 'e' to everywhere, 2003, http://www-1.ibm.com/mediumbusiness/venture_development/emerging/computing.html
10. Holtman M, Mallonee M. New SRI International Study Shows Handheld Computers Can Increase Learning in K-12 Classrooms, SRI International – Online Magazine, 11.11.2002, http://www.sri.com/news/releases/11-11-02.html
11. Wade R. What happens when you hear or see the term "Blended Learning"?, 2005, http://www.elearningeuropa.info/index.php?page=doc&doc_id=4173&doclng=6&menuzone=1
12. Feasey D. Meaning, Meta Data and E-learning. Eyepopping Design. 7.3.2002 http://www.eyepopping.com/pdfs/Meaning_eLearning.pdf
13. Bär H., Trompler C., Rößling G., Mühlhäuser M., Trnková J. Ubiquitous cooperative learning. In: Schubert S, Reusch B, Jesse N, editors. Informatik bewegt. Proceedings of the 32nd annual conference of the GI. Dortmund, GI; 2002. p. 227 [LNI P-19]
14. Malliou E, Savvas S., Sotiriou S., Agogi E. The MoTFAL project—mobile technologies for ad-hoc learning. In: Szücs A, Wagner E, Tsolakidis C, editors. The quality dialogue. Integrating quality cultures in flexible, distance and e-Learning. Proceedings of the EDEN annual conference, Rhodes; 2003. p. 466–71
15. IMS Global Learning Consortium. Meta-data Specification, Version 1.3, 2004, http://www.imsglobal.org/metadata/index.html