Using USB Keys to Promote Mobile Learning

doi:10.3991/ijim.v3s1.843

M. Rosselle[1], D. Leclet[1] and B. Talon[2]

[1] MIS-UPJV Laboratory, Amiens, France

[2] Université Lille Nord de France, Calais, France

Abstract—**M-learning (i.e. mobile learning) is a field of e-learning that provides learners learning environments using mobile technology. In this context, learning can take place anywhere and anytime, in open and distance learning. Depending on the type of technology it may be done through software called nomadic (i.e. prepared to mobility). Among these technologies, there are those composed of digital interfaces and with autonomy of treatment: Smartphone, PDA, calculator and even mp3 key. In this article we propose to take into account storage devices as mobile technologies. Our focus was on the USB key. We present a procedure to test whether a learning environment embarked on a USB key can be described as nomadic or not. This procedure has been tested on a sample of three ILE (Interactive Learning Environment. This approach has allowed us to define criteria of nomadism, criteria which were then included in the design of a synchronous Weblog on USB key.**

Index Terms—**m-learning, mobile blog, nomadism, USB key.**

I. INTRODUCTION

The research described in this article takes place in the framework of m-learning (mobile learning). The definitions of m-learning are manifold [1]. We offer a definition that seems to synthesize the issues conveyed around this concept:

> *"Learning that occurs when the learner is not at a fixed location and/or benefits of mobile technologies. Thus, we believe that m-learning covers ubiquitous learning, which allows the pervasiveness of computers in the learning environment and pervasive learning, that offers a multimodal and multi-channel access to the educational device"* [2].

Thus, issues covered by the m-learning are diverse. How / what to make learn with mobile media? How to adapt software from ILE (Interactive Learning Environment) research domain to mobility? What educational device providing continuity in the learning process can we design? How to adapt content to different equipment and different contexts? What delivery mechanism of content to the learner must be put in place? These are all questions that the m-learning proposes to find answers.

Among the technologies used for mobile learning, we found mainly technologies composed of digital interfaces with a self-treatment (mobile phone, Smartphone, PDA, calculator, game console, MP3 key, etc.) [3][4][5]. They also include technologies such as GPS geolocation or the radio-frequency identification [11][6]. However, to largely cover mobile learning, why not also take into account storage devices like USB key, DVD, or external HD. Thus, is the fact to propose to learners learning environments on such supports in the issues covered by the m-learning? We believe that the issue of the provision of learning environments on such materials can be considered a scientific lock in mobile learning.

Thus, mobile learning is based on the use of mobile software. This software designed for mobility is called "nomad software." The question here is not to study the portability of software on different machines with different operating systems. This issue is studied in the field of operating systems and virtual machines. The issue that interests us is to "transport" software running on a machine of a certain type (e.g. PC running Windows XP) to an equivalent machine type without installation constraint.

In this context, one of our concerns is to provide continuity in the learning process. Indeed, we found, during the evaluation of a pedagogical device called COOLDA-MAETIC, i.e. evaluating the MAETIC method [7] on a malleable platform called COOLDA [8] that students tended to interrupt their learning process. There was a lack of availability of tools. The USB key seemed to be a way to "ensure" continuity during their learning process. However, it was essential to verify the feasibility and ease of transporting learning environments. Thus we decided to define the criteria of nomadism "feasibility".

II. BACKGROUND

In January 2007, a mobile educational device was developed [9] which incorporated a voice server. Although it has shown its technical feasibility, it has not proved to be entirely satisfactory from the point of view of its usability [7]. In January 2008, we experienced an educational device called COOLDA-MAETIC with first-year students of an Associate degree in computer science and with students of a Bachelor Degree in Digital Imaging at the “Littoral Côte d’Opale” University (ULCO). In this experiment, we found that students tended to interrupt their learning process.

The assumption was that interruptions are due to the fact that students do not have a tool configuration, outside the meeting sessions, similar to that available in the classrooms of the university. However, the mobility of working tools could promote work outside the meetings. To make the tools becoming mobile, we propose to board the working environment on a USB key. This offers the possibility to contribute to the group work offline. The USB key is indeed a light, easy and inexpensive way to transport its environment and to work temporarily offline.

In this part, we propose to justify the choice of using a USB key, then we give some definitions and we present a procedure to test whether a piece of software is nomad on a USB key.

A. *The choice of using the USB key*

The choice of using the USB key as a support for mobility is based on the fact that it is a widely distributed and inexpensive. Moreover, its use does not require paying a subscription or a package from an operator or to be permanently connected to the Internet. Furthermore, we all have the opportunity to work on different computers and it would be very convenient to carry with us our software and our work environments, as we carry our documents. The value of a program that runs on a USB key is to allow its owner to work on any computer and retrieve his complete environment, his parameters, like his software. Moreover, this work leaves no trace on the host of the USB key.

For the technician who maintains a multimedia room, the advantage is that he does not have to install software specific to a given class. He installs the standard software of the training and provides connectivity with the Internet if necessary. In UMVF (French speaking Virtual Medical University [12]), the study of the "desk of the student" has shown that the work of technicians to ensure a multimedia room is ready to receive a Medicine course was long and tedious. This requires consulting a repository [10] of materials (e.g. sound card) and software (e.g. pdf reader) and configurations (e.g. 640x480 screen). The use of USB key equipped with nomadic software simplifies that work.

For the researcher or the teacher, the advantage is to provide learners adequate software by distributing keys with adapted and configured software. Furthermore, the key is used to store traces collected during the activity with the software.

The idea to bring the software environment on a USB key is technologically possible. Indeed, USB keys already allow carrying a PC on a USB key. The PC can be equipped with a Linux operating system (like [13] or [14]), which operates independently of the host PC's operating system. There are also solutions such as MojoPac [MojoPac], which can carry on a USB key or iPod some virtual work spaces under Windows XP. The data are separated from the host PC, but it is the latter that works in the background. There are finally nomadic software distributions on USB key using the Windows operating system of the host PC [16] [17] [18].

In this article we will only study software running on PCs using the Windows XP operating system. We will now introduce some definitions to clarify what we mean by mobility offered by a USB key.

B. *The proposal of Some Definitions*

We want to embark on USB key software that is able to operate using only the operating system of the host computer. We call this software: nomad software for USB keys or simply nomad software.

In literature, they are also called portable software. But are these words really equivalent? We offer a definition of these two words: mobile and portable. Furthermore, the literature says that the software running on USB keys is discreet software [18]. We define this term and its relations with the terms nomad and portable.

- Discreet software is software that, when installed or executed, leaves no traces. Under Windows XP, when a software "leaves trace", they are found in the registry or the hard disk in the installation directory or elsewhere (in "c:/documents and setting/software" for example).
- Nomad software is software that runs in a folder and works independently in the latter. So if we move it (e.g. on a USB key), the software is sufficient in itself and only needs the operating system to run. Obviously, if the software writes important data in a folder other than the installation folder or in the registry, and that we want to use the software on another machine, it will not find the right information and not therefore not function properly. So software that comes with a simple zipped file is a better candidate than software that uses an installer because in the latter case, the installer may write to the registry or install files elsewhere that in the installation directory.
- Portable software is software that can run on different machines and on different operating systems. Portable software has a great potential to be nomad because it is less likely to use any subtlety of the operating system of the machine on which it was developed. Free software is good candidates. Indeed, free software developers like to develop for several platforms (Linux, Mac, and Windows). However, a nomad software is not necessarily a portable software and vice versa.

In the context of our work, we talk about nomad software. However, we will not use the portable term that describes a software set, that intersection with the nomad software set is not empty.

C. *The Proposal of a Test Procedure*

In our study, our wish is to exploit existing software in a context of nomadism. We therefore propose to test the software on the basis of feasibility, to know whether they are nomadic or not. In the study, the Procedure Test (PT) we propose, declares "*this software is nomadic*" or "*this software is not nomadic because....*". This procedure does not make it nomad.

Checking nomadism involves verification of a number of constraints. In the previous part, we highlighted the link between discreet software and nomad software. We will therefore in a first time, verify that the tested software is discreet. If the software is not discreet and stores vital information on the hard disk or in the registry, we can stop. It will not be nomad. Once the discretion is verified, we'll check nomadism. The test of nomadism does not alter the original software, i.e. no line of its source code is affected. Finally, remember that the proposed procedure applies to applications running under Windows XP.

1. *Checking discretion*. We check the discretion of an application during its installation and during its use. For the installation, we scan (with RegShot [19]) the registry first and the tree beginning at "C: \ Windows," then we install the software and we scan the registry and tree above a second time. The comparison of the two scans can identify the keys that have been added, changed or more rarely deleted. When software has no installation program but is unpack from a single file (e.g. zip file), this analysis phase becomes unnecessary.

For the use, we begin to move the installation directory of software on another location on the installation disk. We then scan the registry for the first time and the tree starting at "C:\Windows" and beginning to "C:\Documents and Settings". We could scan the entire installation disk, but it is not profitable because, in general, when software writes elsewhere that in its installation directory, it writes in "Documents and Settings". Then we run the application with as many manipulations as possible, especially changing preferences when possible. The manipulations depend on the tested software. Wherever possible, we do a full type exercise with the software. Then we scan again the registry and trees mentioned. A comparison of two scans gives us the number of keys added or changed, the number of files added or changed and the number of files created. This way, we check the discretion of the application during its use.

The most delicate work begins. The question is now whether the keys and files created during installation or use are crucial or not. This is a case-by-case study.

1. *Checking nomadism.* Once the software is installed on the hard drive, we check the implementation of nomadism in two steps. The first step consists to move the software in another directory than the installation one. This step provides a check of the discretion of the application during its use at the same time we check if the software is running normally and does not use paths and logical units depending on the installation directory. The second step is to copy the installation directory on a USB key. Then we test the software on a computer other than the one on which we made the installation. We can be sure that if the software uses keys from the registry database or files stored on the first computer, it will not find them and therefore it will not be able to run normally.

III. APPLICATION OF THE TEST PROCEDURE ON A SAMPLE OF THREE ILE

In Interactive Learning Environment (ILE) domain, there are many applications, based on Windows, that could be adapted to run on a USB key in an appropriated use context. We conducted an exploratory study to test the mobility of these systems, chosen because they were easily downloadable. Our choice was -Cabri Geometry, Aplusix and Pépite. Our goal was to validate experimentally the feasibility of the test procedure and the "conformity" of our criteria. We have therefore, for each ILE, checked the discretion and the nomadism.

A. Cabri-Geometry

Cabri-Geometry is software to draw geometric figures and manipulate them dynamically. Cabri has already been adapted for a calculator which is a mobile device. But does Cabri already fulfill the conditions to run on a USB key? We tested Cabri II+ version 1.4.2 downloadable at [20]

1) Checking discretion

The table below shows the total number of modifications (keys and files) during the installation phase on a Windows XP and during the use phase once the installation directory moved.

Table 1 shows the number of changes (keys, values etc. on lines) during installation phase of the software (center column) or during the use phase (right column). This table shows that Cabri Geometry is not very discreet. But are the added or updated data sensitive? We will not detail here all changes, but will just comment some of them.

TABLE I.
DISCRETION OF CABRI-GEOMETRY

	installation phase	**use phase**
Added Keys	146	0
Added values	286	4
Modified values	6	8
Added files	1	1
Updated files	11	21
Added folder	0	2
Total	**450**	**37**

During the installation phase, Cabri-Geometry creates keys to identify MIME types - Multipurpose Internet Mail Extensions - (HKLM\SOFTWARE\Classes\MIME\Database\Content Type\application/Cabri II Plus). It is not blocking as MIME associations can be made later on another computer if necessary. It amends the key installations of software. This is more inconvenient because it is likely to consult later this information (e.g. HKLM\SOFTWARE\Classes\ Installer\Products\CA6F080C32623B2 ... 6D7485\ProductName: "Cabri II Plus 1.4.2).

During the use phase, Cabri creates a folder in "Documents and Settings\All Users\Application Data\Cabrilog. This is typically the kind of creation that can harm the nomadism. We can notice that the total number of changes during the use phase is relatively low. This is encouraging for the future.

2) Checking nomadism

We moved the installation directory of Cabri-Geometry on a USB drive and launched it on a post on which the installation had not taken place. At the launch of the application, we had an error message calling for the product installation. However, once this error message taken into account, the software is still usable in demonstration mode. A message appears on screen stating that the software is not usable in the classroom. But that does not stop to create a figure and handle it

In conclusion, we can say that Cabri-Geometry is not very discreet. However, it operates in degraded mode (demonstration mode) when it is moved on a USB key. We can therefore consider it is nomad. In any case, the authors of this software does not have a big substantive work to provide a completely nomad version.

B. Aplusix II

Aplusix is a program to learn algebra. It allows to edit algebraic expressions and to manipulate them. We used setupAplusixV1_02_fr_Demo.exe version downloadable at [21].

1) Checking discretion

The table below shows the total number of modifications (keys and files) during the installation phase on a Windows XP and during the use phase.

Table II shows that the software is not very discreet especially in the installation phase. During the use phase, it stores settings and information about the exercises. This can make trouble. Moreover, it creates log files and information about the user in "C:\ Windows\" (6 files) and "C:\Documents and Settings\Users\" (2 files).

TABLE II.
DISCRETION OF APLUSIX

	Installation phase	Use phase
Added Keys	503	5
Added values	706	8
Modified values	11	6
Added files	1	0
Updated files	14	8
Added folder	1	0
Total	**1236**	**27**

2) Checking nomadism

We only tested the students' version. The execution of Aplusix in a directory other than the installation directory but on the same machine and the same logical unit has succeeded. We created a new user (an alias) and produced an exercise in developing algebraic expression. The execution of Aplusix on the USB key on another computer, allow us to find the user previously created and to continue the same exercise as above.

In conclusion, we can say that the two aforementioned executions allow us to say that Aplusix is nomad on a USB key even if he is not especially discreet during the installation phase. Files created on the disk and keys created in the registry do not seem to disturb the correct running of the software on a USB key.

C. Pepite

Pepite is software to learn algebra. Pepite has two applications: PepiTest and PepiProfil. PepiTest allows the learner to test their knowledge of algebra. Once the test and data saved, PepiProfil can analyze the results of the learner and generate a cognitive profile that the teacher can adapt for learning. Pepite is available at the following address: [Pepite]

1) Checking discretion

The table below shows the total number of modifications (keys and files) during the installation phase of Pepite on a Windows XP and during the use phase of PepiTest

Table III shows that the software is fairly discreet during the installation phase. The values and modified files blockers do not seem to block nomadism. During the use phase, many more values and files are added, modified or created. In particular, 12 files in "C:\Windows" (including log files) and the file containing data for the user "C:\Documents and Settings\User\ntuser.dat.LOG" are modified. This can make trouble.

TABLE III.
DISCRETION OF PÉPITE

	Installation Phase	Use Phase
Added Keys	0	41
Deleted Values	0	1
Added Values	0	41
Updated Values	4	17
Deleted Files	0	2
Updated Files	5	13
Total	**9**	**115**

2) Checking nomadism

Running pepiTest, the student' version of Pepite, in a directory "C:\" other than the installation directory, was conducted smoothly. Running pepiTest and pepiProfil on the USB key on another computer running on XP went well. However, during the execution of pepiTest, the program tries to install a component of Windows Office Professional. A message then calls the introduction of the Office CD. We just click cancel. The messages are annoying but do not seem to disturb the normal functioning of PepiTest, which remembers exercises done.

In conclusion we can say that Pepite is nomadic for the USB key (even if we tackle this problem inserting the Office CD). In addition, Pepite is pretty discreet during the installation phase and a little less during the use phase.

This exploratory experiment allowed us, on the one hand, to make a first validation of our test procedure and to test its feasibility. On the other hand, it allowed us to identify two criteria: the discretion and nomadism. Indeed, for software to be discreet, it must not use the registry to store information. Similarly, it should not create file or folder outside its installation directory. To be nomadic, it must operate alone. To do this, it does not seek specific functions of the operating system of the host computer. It does not seek specific components of the system as the Java virtual machine or framework.net. Finally, it should not make use of the software installed on your host (such as ACCESS).

It is on these two criteria that we tested nomadism on USB key of a piece of software widely used today by many users: the Weblog.

IV. A SYNCHRONISABLE BLOG NOMAD ON USB KEY

Remember that one of our concerns is to offer students continuity in the learning process. Indeed, when assessing the educational device named COOLDA-MAETIC, we noticed that students working in group to a project, tended to interrupt their learning process between sessions attendance. We propose then to board their working environment on a USB key and to offer them the opportunity to contribute to the group offline. We want them to include ways to edit the logbook of their project (which is carried by holding a Weblog) offline via a USB key

We have designed a nomad synchronisable Weblog: nomad because our goal is that it works on a USB key and synchronisable to provide synchronization of offline blog with the online blog.

As a first step, we conducted a study of existing technologies. We studied software for Weblog editing (blogger) that offer both a download feature (recovery of a blog to a USB key) and an upload feature (update to the blog site changed on the USB key). Offline Weblog technology requiring an offline customer, we also studied two clients: SharpMT [23] and Wbloggar [24].

In a second step, we designed a synchronisable nomad Weblog into two parts: a server and a client. The client is Windows software. It allows the data backup in a SQLite database and the content creation, while being most nomads as possible. The client is developed in C + + to avoid installing the Java virtual machine or the framework .Net on the USB key. The customer can delete, update or see posts or comments. It can also create posts. The blog server-classic is a website with a remote HTTP/XML interface, which returns responses in XML format and allows to download or to send the blog content from a client. It is written in PHP/MySQL.

We then applied the Test Procedure (TP), explained earlier, to verify the discretion and nomadism of the synchronisable Weblog. Since the server part of the Weblog is a website, we have not had to test its nomadism. However, the client Weblog is a program written in C + + which can both run on a PC with Windows XP or on a USB key connected to a host PC with Windows XP operating system. We then tested the nomadic client Weblog.

Finally, we tested the modification and content creation on the key and on the site and the synchronisation of the key and the site. We have now to test the client and server in ecological environment.

V. CONCLUSIONS

The focus of this article was on the choice of USB key as mobile technology. Thus, after introducing the context of our research, justified the choice of USB key, supported by some definitions, we proposed a procedure to test whether a learning environment embarked on a USB key can be described as nomadic or not. This procedure has been tested as part of a study conducted on a sample of three ILE. This approach has allowed us to define feasibility criteria, criteria which were then included in the design of a synchronisable nomad Weblog on USB key: i.e. running a blog offline on the USB key and synchronizing it with the weblog site.

Our goal now is to conduct an experiment of this synchronisable Weblog with an audience of students at the ULCO in the year 2009. The trends of this work are nomadisation of the working environment COOLDA-MAETIC on USB key. The COOLDA-MAETIC working environment is built around a development environment (Eclipse). It has a chat, a CVS, an office suite (open Office). We will add our synchronisable Weblog to this environment. To test our environment, we will apply our testing procedure. We have already seen that Eclipse is nomad on USB key. The office suite is already nomad and available on the Framakey site [18]. We will thus check the nomadisation of the CVS and of the chat to set up the experiment. The results of this experiment will be the subject of future communications.

REFERENCES

[1] K. Masters, “M-learning: how much of what has been diffused? A systematic literature review”, ED’MEDIA 2008, 5790-5796

[2] C. Quénu-Joiron, D. Leclet, B. Talon. “Conception de dispositifs pédagogiques intégrant la mobilité”. Atelier Apprentissage Mobile. Associé à la conférence EIAH 2007. p17-20. Lausanne, Suisse. 26 juin 2007.

[3] E. Webb, G. Cavanagh “How Mobile is your Podcast?” ED’MEDIA 2008, 3954-3958

[4] C. Salis, M. Ambu, “MOsKA, Mobile Organized Knowledge Access for Science: Astronomy and Renewable Energies Videos for Mobile Phone Delivery”, ED’MEDIA 2008, 4388-4393

[5] J. Arreymbi, E. Agbor, M. Dastbaz, “Mobile-Education - A paradigm shift with Technology”, ED’MEDIA 2008, 5114-5122

[6] K.-J. Huang, T.-C. Liu, S. Graf, Y.-C. Lin, “Embedding mobile technology to outdoor natural science learning based on the 7E learning cycle”, ED’MEDIA 2008, 2082-2086

[7] D. Leclet, and B. Talon, “Assessment of a Method for Designing E-Learning Devices”, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, ED-MEDIA 2008, AACE/ Springer-Verlag (Ed.), Vienna, Austria, June 30 - July, p 1-8, 2008.

[8] A. Lewandowski, G. Bourguin, “A New Framework for the Support of Software Development Cooperative Activities”, Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS'06), Paphos, Cyprus, 23-27 May, 2006, ISBN: 972-8865-43-0, INSTICC Press, pp. 36-43, 2006.

[9] D. Leclet, E. Leprêtre, Y. Peter, C. Quénu-Joiron, B. Talon, T. Vantroys. "Améliorer un dispositif pédagogique par l’intégration de nouveaux canaux de communication" Environnements Interactifs d’Apprentissage Humain, EIAH 2007. p347-357.

[10] M. Rosselle, P. Gillois, J. Morinet-Lambert et F. Kohler. » Installing the Student’s Computer to Access a Virtual Medicine University”. International Symposium TICE’2002, Technologies of Information and Communication in Education for Engineers and Industry. INSA, Lyon, France, 11-15 nov. 2002.

[11] radio frequency identification http://en.wikipedia.org/wiki/RFID last visited on 14/11/2008.

[12] Université Médicale Virtuelle Francophone http://www.umvf.prd.fr/ last visited on 14/11/2008.

[13] Linux on USB key www.knopper.net/knoppix/index-en.html last visited on 14/11/2008.

[14] Linux on USB key www.pendrivelinux.com last visited on 14/11/2008.

[15] virtual workspace www.mojopac.com last visited on 14/11/2008.

[16] Windows software on USB key http://portableapps.com last visited on 14/11/2008.

[17] Windows software on USB key www.winpenpack.com last visited on 14/11/2008.

[18] Windows software on USB key www.framakey.org/En/Index last visited on 14/11/2008.

[19] Software to scan registry and file tree sourceforge.net/projects/regshot last visited on 14/11/2008.

[20] Cabri http://www.cabri.com/cabri-2-plus.html last visited on 14/11/2008.

[21] APLUSIX II: Algebra Learning Assistant http://aplusix.imag.fr/en/index.html last visited on 14/11/2008.

[22] Pépite software (in French) http://pepite.univ-lemans.fr/ last visited on 14/11/2008.

[23] blog client http://www.codeplex.com/sharpmt/ last visited on 14/11/2008.

[24] blog client http://www.wbloggar.com/ last visited on 14/11/2008.

AUTHORS

M. Rosselle is a member of the MIS-UPJV Laboratory, 33 rue St Leu, 80039 AMIENS CEDEX 1, France (e-mail: marilyne.rosselle@u-picardie.fr).

D. Leclet is a member of the MIS-UPJV Laboratory, 33 rue St Leu, 80039 AMIENS CEDEX 1, France (e-mail: dominique.leclet@u-picardie.fr).

B. Talon is a member of the LIL-ULCO Laboratory, Maison de la recherche Blaise Pascal, BP 719, 62228 CALAIS CEDEX, France (e-mail: talon@iutcalais.univ-littoral.fr).

This article was modified from a presentation at the International Conference on Interactive Mobile and Computer Aided Learning (IMCL2009) in Amman, Jordan, April 2009. Submitted, February 20, 2009. Published as resubmitted by the author(s) on July 20, 2009.