First Experiences with OLPC in European Classrooms

MARTIN EBNER
Graz University of Technology, Austria
martin.ebner@tugraz.at

JOHANNES DORFINGER
University of Teacher Education, Styria, Austria
Johannes.Dorfinger@phst.at

WALThER NEUPER
Graz University of Technology, Austria
neuper@ist.tugraz.at

For many years now E-learning researchers have been discussing the use of laptops in educational settings. The One Laptop per Child project, (OLPC project), has been used in developing countries since the year 2002 to avoid the increase of the digital divide between those with ready access to technology and those with limited access. Austria has become one of the first countries in the European Union (EU) to start an OLPC-project of its own. The focus of the project was on the use of digital devices in education at elementary school level. Accompanied by a solid research team, which brought teachers, e-learning experts and a software developer together, a first attempt was established. This article will describe the preliminary work and the first real life setting of this project, concluding with the experiences of the whole research group. It will also summarize a recommendation for the transfer of the project to developing countries.

THE DIGITAL GAP

Digital technologies have begun to pervade our daily lives through the omnipresence of the internet via many different devices. As Mark Weiser stated, “the most profound technologies are those who disappear” (Weiser,
1991) and he also described for the first time the idea of pervasive computing. The emphasis and availability of the World Wide Web as well as computer technologies have lead to a dramatic change as to how mankind is working and dealing with digital data. These changes have also had a great impact on teaching and learning behaviors. After Tim O’Reilly introduced Web 2.0 in 2004 (O’Reilly, 2005), Stephen Downes described the use of Web 2.0 technologies for education as ‘E-learning 2.0’ (Downes, 2005). The Web has become a community of loosely connected participants who not only share their data but also preferentially their knowledge and interests. Communication and collaboration have become key factors (Tuchinda et al., 2008).

By understanding increasing connectivity through different devices like this, the famous expression A3 (anytime, anywhere, anybody) has reached a new dimension. Enhancing traditional as well as online teaching and learning standards by universal devices is called: u-learning. Zhan & Jin (Zhan & Jin, 2005) defined u-learning as a function of different parameters:

\[
u\text{-}\text{learning} = \{u\text{-}environment, u\text{-}contents, u\text{-}behavior, u\text{-}interface, u\text{-}service\}\]

It can also be stated that the switch to u-learning needs entirely new didactical approaches (Holzinger et al., 2005) (Ebner & Schiefner, 2008). On the one hand it must be taken into account that digital technologies can be used in a very explicit way. On the other hand one must also consider how e-literacy for people can be guaranteed for those that are involved in teaching and learning. Gilster (Gilster, 1997) defined Digital Literacy as the ability to understand, evaluate and integrate information in a variety of formats delivered by computer. Eshet-Alkalai (Eshet-Alkalai, 2004) extended this definition and provided a conceptual framework by splitting it into several cognitive skills: Photo-visual, reproduction, branching, information and socio-emotional literacy.

However, in the digital world of tomorrow educational experts have to ensure that children are educated in terms of digital literacy. They must be appropriately prepared for a broad digital environment. Furthermore problems like the Digital Divide must be taken into account as described by the Association for Progressive Communication Organization as “the increasing gap between those who have and those who do not have access to information and communication technologies, access to content that benefits them socially and economically, skills to take advantage of ICT services, and the ability to afford to pay for digital services.” (Web 7).

The non-profitable organization OLPC must be pointed out at this moment as an important contribution/contributor in this context. The mission
of the project is “to empower the children of developing countries to learn by providing one connected laptop to every school-age child” (Web 1). Of course the major focus of the project is to bring digital technology to developing countries, but the use of low-costs laptops in European classrooms, especially for children, is of high scientific interest. Several projects have already been carried out (Martinazzo et al., 2008) (Fichemann et al., 2008) (Franco et al., 2008) focusing on South America, in particular Brazil.

In this paper we will describe the first steps using XO (which is the name of the low-costs laptops) in European classrooms.

**OLPC INITIATIVE / RESEARCH PROJECT IN AUSTRIA**

**OLPC Initiative**

Professor Nicholas Negroponte founded the OLPC initiative in 2002 and aimed to “provide a means for learning, self-expression, and exploration to the nearly two billion children of the developing world with little or no access to education” (Web 1). The main aim of the project is to improve education in countries that have less technology by bringing robust and usable digital mobile devices. According to the digital divide (Nielsen, 2006), the project focuses on addressing the economic divide, bringing computers to children of developing countries and thereby improving and raising their digital literacy so that it is more similar to those of industrialized countries.

The concept of the OLPC project is a very simple but powerful one: For each XO computer bought by anyone from an industrialized country a second one without hardware and software issues is sent for free to a developing country. In other words the slogan might be “buy two, get one and help anyone who needs it."

It is easy to imagine that there have been and still are special requirements in the development of the hardware and software of such an XO as a result of lacking environment settings in the developing countries as well as lacking monetary issues. For example, developed software must be offered as an open-sourced product, which is available worldwide. The XO’s operating system, called Sugar, is based on Linux and all programs use the programming language python.

Furthermore, to enhance the community, partners all over the world have been established. In Austria the non-profit organization OLPC Austria (Web 2) cares about spreading this idea to educational institutes. There is of course a close relationship between this association and the project described in this paper.
OLPC Initiative at the University of Teacher Education in Graz

The OLPC project at the University of Teacher Education in Styria (PHST) is the very first OLPC school project in Austria, and furthermore the first within the whole EU. The project helps us to teach the children in a very new way. The involved project partners had to deal with completely new teaching methods and to evaluate their actions constantly.

The project started following these basic decisions:

- The Austrian Federal Ministry for Education, Arts and Culture (bm:ukk) called for participation on an OLPC-project. Four primary school classes were to be equipped with 25 XOs each.
- The hardware was sponsored free of charge by OLPC Austria.
- Because four classes should participate, there should have been a special training program for the teachers of the concerned classrooms. Special software for the target group should be developed before the beginning of the next school term.
- The research results should affect the studies of all forthcoming teacher-students at PHST.

In November 2008 the PHST started the current project with 25 XOs. The project class is a first class at the Praxisvolksschule (PVS) of the PHST. This PVS is an elementary school for pupils between 6 to 10 years. As a part of the PHST the PVS tries to keep up-to-date in modern teaching methods and technical equipment. Our student teachers have to train and practice their teaching skills during their whole studies to perfect their teaching methods. The parents had to apply their children for this special project class with a written statement. Although the pupils are normal average children, we thought it was important for the parents to know, that the teaching methods in the project class will differ from normal classes.

The class consists of 25 pupils (10 boys and 15 girls) at the age of six. The children will stay at this school for four years. Throughout these four years they will use their XO laptops. As this is an all-day class, the pupils stay in school until 3 p.m., which includes lunchtime and doing homework. A female and a male teacher alternately work in the class to balance gender representation. As the pupils of the project class complete all their homework at school, XO laptops are not to be taken home. The laptops remain at school and are only used when needed. In the normal school day, XOs are in use for several 30 minutes blocks. These blocks allow the pupils to practice the subject matter and work out new information in a connected and collaborative surrounding. The XO – as an additional tool – became an important part for working on the subject matters in a new exciting way. Figure 1 shows pupils working with the XOs in classroom.
Research Team and Role of Partners

It appeared quickly that in order for the project to be successful, further scientific partners had to be involved. Activities using the XO in the standard distribution covered a wide range of educational settings. Nevertheless, we expected some gaps in covering the elementary school curriculum (which addresses illiteracy at the beginning) and anticipated respective development efforts.

As a result collaboration with Graz University of Technology (TU Graz) has been set up: The collaboration included the Department Social Learning at TU Graz (SL) and a team at the Institute for Software Technology at TU Graz (IST). The roles of the partners are multifaceted in order to ensure a highly valuable research output:

- SL: To gain experiences about e-learning in general and support in instructional design of new software as well as didactical approaches in using it. Furthermore the department supervises a project about usability issues of the software – how are children using the special software, and what is the usability of the interface like – as well as about the user requirements - teachers as well as children – to achieve the intended tasks.
• IST: The main research areas of the IST are formal methods for software development, and the main teaching areas are the basics for such research; in addition to that the lecture *Didactics of Informatics* is also hosted by the Institute. It was consequently straightforward to establish a development team under the supervision of the IST. The team collaborated with the SL with respect to user requirements engineering on the one hand, and with OLPC Austria as to the software requirements and architecture on the other hand.

• PHST: Two Bachelor-Students of PHST are carrying out their theses by monitoring the progress of the children in terms of their mathematical and literacy skills. Four tests in seven months are planned in order to establish a pre/post-test experimental control group design. In addition, the actual working time (in contrast to management time) in the classrooms of the XO laptops will be measured as well. The importance to distinguish between different times is well found in technical problems, which distract children from their core learning goals. In this particular case pupils of the project class as well as pupils of the control class would have had different practicing times.

• Non-profit organization OLPC Austria: OLPC Austria supports the project with technical know-how and helps to exchange experiences with the global OLPC network. Our results are returned to the main OLPC developer team in order to meet the standards for the worldwide OLPC project.

Developments in the First Project Year

We experienced some minor and major problems during the first project year, which will be described and discussed in more detail at a later stage. The problems varied from instable WLAN connections and missing applications as well as very difficult beamer connectivity. Although we were able to solve most of those problems we found it very difficult for a “normal” school to deal with such a large amount of problems. Due to the close cooperation with our project partners the project could finally carry on.

But it is not the problem list that makes this project remarkable. It’s actually much more the list of benefits that makes us work even harder. It is very obvious the children really benefitted greatly by working with the XOs. They became very competent in problem solving strategies. When we first recognized, that the battery performance was far below our expectations, we thought that the children would be very desperate if they lose some data, or probably even a gaming high score. But in fact they arranged themselves with the situation. They recognized the low battery level, plugged in their machines and carried on working at the edge of the classroom without any instruction from the teachers. From time to time, this ended up being a rath-
er curious scenario as most of the pupils worked on the edge of the classroom, but it is a good way of problem solving. We also recognized that they used such problem solving strategies with the software as well. At the beginning of the project they kept us busy helping them solving software problems. During the first project year the need for such support diminished. The reasons for that were: firstly they got used to working with the computers. As a result fewer problems occurred. Secondly they invested more energy to solve the problems themselves. They developed an ambition to handle their problems without the help of adults. If they could not solve a problem themselves, they would firstly ask a fellow pupil. And although enough questions were still addressed to the teachers, we recognized that there was a very strong community in the class. Therefore a study has been initiated to measure the social capital and the development of creativity within the project class. The bm:ukk assigned the Büro für die Organisation angewandter Sozialforschung (BOAS) (Web 3) to complete the study. The results were amazing! They show that the social capital in the project class was much higher than that of in a comparable class.

For the project class these are important results. Prof. Gehmacher from the BOAS who carried out the study summarized the results as follows:

- The school won attraction for the project class.
- The Social capital (circle of friends) increased strongly in the project class.
- Family bonds strengthened in the project class.
- Dream jobs were seen more objectively in the project class.
- However computer games displaced drawing and playing music as activities.

So the project created a bond of companionability in the class. The pupils could/can identify themselves as the pupils of the laptop class. They belong together. This in turn turns each pupil into something special; they have something to show off, can tell stories and adults listen to them. And of course they enjoy this.

The second aspect that occurred was that the difficulties in learning languages and writing were bigger than we expected. We knew that the keyboard on the XOs has an English design, which was not the cause of the biggest problem. But although one can switch to the German language, some parts of the system are optimized for English. The audio output of the application called speak was also faulty. The teachers actually thought that the letters were actually pronounced as they are written rather than with vowels. An m for example is just an m, not am em. The software on the other hand pronounced the German consonants with a vowel added. Of course this was
very confusing for the pupils. The teachers kept on saying *m*, the computer said *em*. We therefore decided to stop learning languages with the XOs and rather concentrate on the scientific disciplines of the curriculum. The generation of the *Reckon Primer* application described in the next chapter was the logical consequence of this decision.

Of course language had to be taught with or without the usage of the XO. Although there was no proper software available for learning the German language, we found some ways to exercise the learned letters. The decision was to use the e-learning platform Moodle to provide some exercises and tasks for the pupils.

![Image of XO with Moodle interface](image)

*Figure 2. Usage of Moodle on a XO.*

In addition to Moodle they also practiced the letters they had learnt by typing on the keyboard. By the end of the year the pupils already managed to send an email to their parents (Figure 3), which was seen as a great achievement by both the parents and the children!

In this way linguistic and mathematic competences were built with the direct or indirect help of the XO. We should also not forget the side effects of using digital tools at such an early age. The effects of collateral learning are important for cultural techniques of the twenty-first century. It shall also help to enhance sentences like the one of Marc Prensky (Prensky, 2006, p. 4), from: “Kids learn more positive, useful things for their future from video
games than they learn in school," to "Kids learn just as many positive, useful things for their future from video games as they learn in school."

In this way linguistic and mathematic competences were built with the direct or indirect help of the XO. We should also not forget the side effects of using digital tools at such an early age. The effects of collateral learning are important for cultural techniques of the twenty-first century. It shall also help to enhance sentences like the one of Marc Prensky (Prensky, 2006, p. 4), from: “Kids learn more positive, useful things for their future from video games than they learn in school,” to “Kids learn just as many positive, useful things for their future from video games as they learn in school.”

**RESEARCH STUDY – THE FIRST PROTOTYPE ENTITLED RECKON PRIMER**

A priority list of development tasks was agreed on between the project partners, with the highest point being to develop software supporting systematic exercising in calculating (Web 5). The idea was to develop an application which can support the learning and teaching of basic arithmetic operations. The main challenge was creating an appropriate tool for children aged 6-7 years old. Until this present moment in time we have named the program Reckon Primer.
Experimental Setting at the University of Teacher Education in Styria

The first year of primary education is the beginning of the Austrian school system. The children come right out from the kindergarten where they are predominantly used to playing and learning voluntarily. Consequently most exercises are still based upon games in primary schools. Teachers have to be very creative to find ways in which to make learning exciting. Of course the XO also has to fit into this category. Collaborative writing and drawing cultivate classic educational goals as well as social skills. But scientific subjects also have to be covered in these early years. One of them is to learn the four basic mathematical operations; in this subject the project partners asked for software support.

General didactical Requirements for Teaching Arithmetic

The didactics of teaching basic mathematical operations is well developed in Austrian primary education. As a result these standards in didactical principles had to be transferred to educational software in order to enhance traditional face-to-face education.

The standard sequence of learning mathematical operations, beginning with addition and subtraction, starts off by going through the following standard sequences:

• (A) From zero to ten, with lots of tactile and visual experiences through working with small numbers of chestnuts, stones and jumps etc
• (B) From zero to thirty, but without crossing the ten barriers
• (C) Carry over the ten barrier, etc.

This standard sequence shows immediately that the use of assisting software is heavily restricted in these stages of learning; the weaker and the less intuitive experience with numbers is, the more sophisticated appropriate software must be. The role of digital devices, however, becomes more and more of a practical role. Furthermore learning to handle numbers and characters and designing numbers is not only due to exercising, but also to processes of mental maturation; the strength of computers can be the possible individualization of learning behaviors.*NASTY PARAGRAPH

The initial part of the above standard sequences (A) (B) and (C) is already long enough to point out the interdependencies which have to be modeled in the software in a way that each individual pace of learning is supported. An example of sequence (C) is presented here:

(1): $7 + 5 = \_\_.\_

This calculation can only be accomplished, if the learner finds out a way to split 5:

(2): $3 + 2$. 


In order to split 5 it is necessary to complement 7 to 10 by 3. Next 3 is subtracted from 5 which equals 2. After that 2 only needs to be written to the unit position of the result yielding as summary from (1) and (2)

(3): \(7 + 3 + 2 = 12\)

Therefore, in order to accomplish sequence (C), specific learning outcomes of sequence (A) are necessary, for example exercises such as \(7 + \_ = 10\), \(5 = \_ + 2\) or \(5 = 3 + \_\). Austrian experiences in teaching basic arithmetic operations show which prerequisites are required for which sequences, how learning steps of each sequences are related, and which exercises are necessary in order to get a maximum of success with a minimum of effort.

If we take a closer look at the provided example it is easily noticeable, that the number 2 occurs twice, in line (2) and line (3). This fact (see Figure 4) can be used to create an analogous exercise leading to learning results, which are more stable and also more easily reusable in other situations:

This exercise concerns the same calculation as before but in a slightly different way:

(4): \(7 + \_ = 12\)

This variation enforces the preparatory exercises from above within the steps as follows:

- (a) Complement 7 to 10 by 3, that is calculate \(7 + \_ = 10\) results 3.
- (b) Copy the unit position of the result 12 to the intermediate calculation started in (a), \(3 + 2 = \_
- (c) Complete the intermediate calculation \(3 + 2 = \_\) results 5.

This variation is not more complicated, quite the contrary: the steps (1) to (3) lead learners to working out to skip (2), which enforces her/him to learn another 36 calculations \((7+1, 7+2, 7+3, \ldots)\) by heart; while the steps (a) to (c) reuse what has already been learned in phase (A): in addition to the area of \(0\) to \(10\).

Or the other way round: if sequence (A) neglects experience and flexibility and superimposes it with memorizing the \(10 \times 10 = 100\) calculations
below 10, this will lead to memorizing another 36 calculations passing the
ten-barrier (and so on). This early phase of learning fills the specification of
a talented mathematician: instead of memorizing calculations learnt by heart
in many cases a talented mathematician flexibly rearranges a calculation to
make it simpler, and then accomplishes the simplified calculation.

It is easy to imagine that all necessary sequences (A) (B) and (C) for
learning basic arithmetic operations should be covered by one software
product which adopts these teaching requirements in an understandable
fashion. The used application must therefore be highly flexible in order to
cope with differences between habits of calculations in different countries.
The Austrian sequences of learning phases are optimized toward the kind of
written division as usually taught in German speaking countries. Thus these
sequences need to be changed in order to be optimal for division as taught
in English speaking countries or others.

The Software Development Cycle

The XO’s user-interface Sugar follows clearly cut pedagogical concepts
(Web 4). To follow these concepts while extending the basic set of activities
was therefore the central issue for development at TU Graz. In solving this
issue, the most interesting experience turned out to be to concern communi-
cative details of the development process rather than technical details. We
therefore restrict the description below to communicative details.

Observations About Competent “Ordering Customers”

The development process started with a requirements inventory (Bjørner,
2006). The initial meetings with the teachers at PHST pointed out, that there
would not be a final product according to some requirements, but that any
software product would not succeed without accompanied support struc-
tures.

The conclusions from the developers’ point of view were:

• The teachers were ICT-literates, who had received teaching in didactics
  of e-learning during their education or during further education. They had
  clear opinions about what kind of classroom activities could be supported
  using which kind of software tools; they were far-off being satisfied with
  the software contained in the standard OLPC distribution.

• The teachers asked for the software to be clear and of a high standard,
  since the school administration requested a proof that the introduction of
  computers did not reduce the success of the learners.

• Teacher and experts on didactics supported their requirements of the
  software with well-established principles in didactics of mathematics in pri-
  mary education (Schwetz, 2001) and with their own respective long-proven
  experiences.
This therefore meant that all electronic communication which had been established by the project leader (mailing list, blog, OLPC wiki …), were immediately used to fulfill the teachers’ requirements of the software with the (limited) capacities of software development.

**User Requirements for Reckon Primer**

Before the core development of the Reckon Primer started, crucial user requirements were collected by a student of informatics. From a programming perspective users are teachers as well as learners required that:

1. Reckon Primer will be integrated into primary education. In particular, there are a lot of indispensable activities for preparing school children to take advantage from Reckon Primer.

2. Users of Reckon Primer are teachers as well as learners. Teachers will be the main persons to adjust operations and learners should solve it.

3. The typical user is an illiterate person: Main users are school children in the age category of five to six years old who are unable to read or write, or maybe adults who want to learn basic cultural skills. As a result of this fact written text on the screen must be avoided.

4. If a screen is used there should be minimum movement: It is highly confusing for young learners if too many animations are moving around; the screen layout must remain as simple and stable as possible.

5. Numbers input are of same size as on keyboard: A further requirement on a simple layout: even the font used for the numbers must be as similar as possible to the numbers on the keyboard.

6. The four basic arithmetical operations are covered: This includes all necessary calculations by heart, particular preparations for handwritten operations, and the operations themselves are written with paper and pencil. The main software has to provide all possible calculations defined in the appropriate area.

7. The content must be divided into sequences: The following sequences (also described in previous chapters) are suggested to be the initial ones: (A) from zero to ten, with lots of tactile and visual experiences caused by operating with small numbers of chestnuts, stones and jumps etc (B) from zero to thirty, but without crossing the ten barriers (C) carry over the ten barrier, etc.

8. Return key avoided: Typical learners at this age will not be accustomed to usual keyboard usage. Thus interaction must be such that the return key is not needed.

9. Displaying graphics for explanation is optional: As mentioned in user requirement 1, comprehension by grasping things should be com-
pleted before the computer is used for exercising. Graphics might help, but graphics might also distract attention and graphics are therefore optional.

10. Summative feedback is optional: According to user requirement 3 and 4 the desktop must be stable and without text, but advanced learners may also like to take note of their results.

11. Immediate feedback is always displayed: The learner immediately sees which calculation is correct and which is wrong: a cross for wrong and a tick for correct. **Immediate feedback strengthens the learning outcome.**

12. The learner can choose the provided sequences: Learners are able to choose what kind of calculations he/she wants to do (described in user requirement 7), but the choice can be restricted by user requirement 14.

13. The learner can choose the level within a sequence: The level is given by the size of numbers, by one or more operations mixed, by the argument asked (?+3=5, 2+?=5, 2+3=?, 5=2+? ...) and by respective mix.

14. The teacher can preset problem class and level: Teachers can adjust which class of calculations and at what level the learners must solve.

15. Recording of needed time: Time needed by each learner for each assignment must be recorded. Afterwards the teacher will be able to analyze the personal learning success of the pupil.

16. Monitoring: As well as the time factor the learner’s performance should also be recorded. The results of each assignment will be saved and afterwards be given to the teacher to have an overview about the learning process.

**Development Cycles and Division of Labor**

Very soon afterwards the following needs for the structure of the software development process became apparent

**Short Cycles of Development – Feedback:** Since the development efforts started after the XOs were first introduced in class several months’ beforehand, specific software was needed at once. Another good reason for short development cycles was the fact, that software with rather novel features had to be developed – for justification of the novelty see the subsequent paragraph. Just re-engineering existing textbooks or software would not have covered the requirements demanded. Furthermore, novel features needed rapid prototyping and early feedback from the users, from teachers and subsequently from learners.
**A Mediator Role for Didactics Experts:** The requirements gathered from the teachers during the initial meeting needed documentation. Elaboration of these requirements confirmed the well-known fact (Bjørner, 2006), that this task required domain expertise as well as technical expertise. It pointed out that there were hurdles in mutual understanding, and that the two parties needed to develop a common language. Since we didn’t want to distract teachers from their task of personal communication with their pupils, teacher students from PHST and TU Graz were engaged in the following tasks:

- Elaborate the user requirements document: this task coordinated the didactical expertise of the students from PHST with the technical expertise of the students from TU Graz
- Check existing software for the requirements as established, since it made no sense to reinvent the wheel. Such research also collected various ideas to be discussed with respect to future development.
- Provide details for cooperative ranking within the priority list of development tasks: for example the requirement ‘We need exact ‘Austrian Schulschrift’ versus ‘We need general feedback of this specific kind’
- Perform usability-testing (Holzinger, 2005) equipped with the usability kit provided by TU Graz.
- Evaluate pupils’ success, following the test design established by the project leader.

In our experience the teacher students performed well in these comprehensive and laborious tasks and adequate supervision was provided.

**Openness to a Variety of Application Scenarios:** After the first development of feedback cycles some requirements were adapted by the teachers; they were considered to be too specific. This decision came about by reviewing the many variants and alternatives on exercises. Teaching is an individual activity and individualization is a specific strength of computer supported learning. By being aware of the fact that learning is an individual activity oriented process, the resulting design decisions were clear (Bjørner, 2006):

- Parameterize methods as much as possible; in our case for example the generation of numbers for exercises.
- Predefine parameters in respective settings; in our case the kind and frequency of feedback, explanations by graphics, etc.
- Abstract to appropriate data structures; in our case exercises which can be defined, sequenced, edited.

Parameterization, settings and definition of exercises within learning sequences required authoring; thus the generalization of e-learning tools
raised demands on those who used the tools i.e., learners as well as teachers and course designers and this was consequently subject to subsequent subjection. One issue of software design is to provide all users with appropriate authoring tools and furthermore to provide the learners with feedback not only about the actual exercise but also about their progress within their course.

Findings for Development of E-learning Tools

The three points mentioned in the previous section are considered typical for educational software development processes — an interdisciplinary cooperation between developers, experts on didactics and teachers. The active involvement of both i.e., experts on didactics and teachers, is relevant to the production phase of learning and exercising in order to accomplish appropriate adaption of the exercises. Thus the involvement shifts from developers to users during the software life-cycle as follows:

1. Usability can be optimized by short development – feedback cycles. Feedback from the learners is not the only important source of feedback, but also from those who are concerned with preparing, monitoring and evaluating the learning process, that is the teachers.

2. Individualization is only possible by profound abstraction of learning processes, which systematically identifies the possibilities of variation; this abstraction requires intimate cooperation between didactic experts and technical designers.

3. Authoring is the consequence of features for individualization (parameterization, settings, definition and sequencing of exercises). The acceptance of authoring by the teachers in the project team was easily achieved, because it reflected their personal expectations.

4. Model courses for different learning situations should be created as soon as possible. Such different situations might be different abilities of the learners, different teaching styles, different kinds of school organization (all-day or half-day school) etc.

5. Community building among the teachers might originate from the teachers involved in the project. They are allegorically the seed that shall allow their enthusiasm to grow within their colleague. Their experiences and even their visions for further development of the software. We do not advise, however, that the community should be centered on any kind of software; rather it must be more general and concern at least the whole subject matter.

6. A support system should be provided by the school administration, which enables the teachers to cooperate personally and to exchange ex-
experiences, which methodizes the learning sequences for individual adoption, and which installs respective Web 2.0 facilities.

Point 4, 5 and 6 have not been realized yet within the actual project. These points state the challenge for educational administrators to transfer the involvement and enthusiasm of teachers in the development phase into the production phase. In the following discussion this challenge is even extended to developing countries.

**DISCUSSION**

**Experiences From the Real-Life Adoption**

Unfortunately only one of the four project classes was approved, unlike the intended collaboration plans. Another problem was the delivery of the XOs: they were delivered to the schools eight weeks after the start of the school-term. Therefore the hardware as well as software could not have been tested in order to brief teachers on their use. As a result of this teachers fell back into established teaching methods and the use of the XOs were reduced to a further additional tool for education. Some technical difficulties made it even harder for the classroom teachers to use the XOs during the lessons:

- **Performance:** The most obvious problem within the real life setting was the rather bad performance of the OLPC laptop. It took a very long time booting the devices and children become nervous and impatient. This results in a clicking-reaction which leads to shutdown and freezing computers. The situation can be described as a very chaotic one, and teachers became stressed.

- **Design:** Most of the applications on base of the Sugar operating system work with written buttons; for six years old children it is hard to use them without guidance. For example if they stop an activity, they have to press a Stop button. The main design was built to be easily understandable, even without reading. But if they made any changes without saving, the application asks whether to stop with or without saving. For children who even cannot read a single German word yet, it is impossible to answer this question.

- **Beamer:** A beamer would be very useful for the teacher in classroom, but there’s no way to connect one to an XO. A classroom server had to be implemented on a standard PC. The Sugar user interface can be beamed easily.

- **Connectivity:** The mesh network, which should connect different XOs
just in time didn’t seem to be very stable. If all the pupils were connected at the same time, they temporarily got disconnected. A reconnection is hard, for they have to read again. Therefore we installed a WLAN router; however this in fact did not help much. The classroom server had to be used again.

- Activities: The applications (activities) on the XO were spread through a wide range of age groups but lacked some basic functionality. But from the beginning it was essential to have software for the training of reading, calculating, writing and typewriting. Such programs were not available yet.

**Experiences of the First Developing Steps**

The first steps of development might be quite typical for educational software:

- A rapid prototype was delivered to the school (PVS) as soon as possible. However, functionalities implemented first have been determined carefully with respect to didactical (not technical) considerations: The learner’s freedom of choice was ensured immediately. In this case the choice concerns the range of numbers (in this case 0 to 5) and the position of the cursor for input (in this case all three possibilities for one calculation, for example $2+3=?$, $2+?=5$, $?+3=5$). There were other choices as well. Choice is indispensable for individualization and for fostering independent learning.

- The above choice was selected as default, because earlier experiences with such software predicted difficulties for some, but not all learners in the XO class. This fact not only motivated learners to adapt their exercises, but also motivated teachers to reflect certain difficulties - and such reflection made teachers state further user requirements.

- The conclusion from the first feedback-cycle at the end of the school year 2008/09 was that the free choice for learners should be overlayed by a list of exercises predefined by the teachers. This list is being implemented by the students at IST during winter term 2009/10 and is still in progress.

Other features have been postponed after the first prototype, since engagement of teachers is definitely more persistent: authoring, course design and support system will be an issue for years.

**Transfer of Schooling Experiences to Other Cultures?**

OLPC’s mission is “to provide a means for learning, self-expression, and exploration” (Web 6), and this mission statement is in line with presently requested pedagogy emphasizing individuality and creativity. Even in scientific education, inquiry-based learning is requested by the Rocard report (European Commission, 2007), which strives for “a renewed pedagogy for the future of Europe."
These pedagogical strategies can be seen as efforts to overcome educational structures remaining from the Industrial Age: schools bound to narrow curricula (in analogy to uniform products delivered by mass production), with specializations in subjects, in age of students, in levels of ability (in analogy to specialization in assembly lines), competitive work ethics etc. These structures and attitudes are considered inappropriate to cope with the challenges in the Information Age. If pedagogical strategies were transferred, enhanced by technology to cultures in a still preindustrial socio-economic state, the question consequently arises: Are educational attitudes symptomatic for the Information Age appropriate for these societies?

With this question in mind it can be stated, that this research study decided to start with a software prototype, which can be allocated to drill-and-practice: Does the project partner’s decision for this kind of software indicate their predilection for outworn educational attitudes symptomatic for the bygone Industrial Age? No, it is rather a well proofed concept on a new device to enhance existing strategies. The main goal is to teach and learn mathematics, a science which is still the driving force for many developments worldwide, as well as numbers and arithmetic operations belong to the first steps towards mathematics. The software prototype supports prerequisites for precise thinking as well as adaptively in its instructional styles.

**Recommendations for Transfer to Developing Countries**

As just mentioned, the transfer of products and techniques into other socio-economic contexts and in particular the transfer of educational material to other cultural situations needs caution and careful cooperation. This fact is well reflected by the XO’s concept of basic (software supported) activities, which are open for varieties of usage and for adaption to the users’ needs.

The OLPC project, however, strives for highly elaborated methods of didactics for the basic arithmetic operations. Nevertheless, we do not recommend a top-down transfer via ministries and universities within official foreign aid; rather, we suggest trusting the democratic potential of ICT and following a bottom-up approach:

- Rely on the facts, that numbers are ubiquitous, that operating with numbers is an indispensable civilization technique (independent from socio-cultural affairs), that calculating by heart needs exercising all around the world. Thus we have:
  - All prerequisites for continued acceptance
  - Good reasons for mutual understanding on this topic from international contacts
• Occasions for school twinning, not only about this topic (as it may be fun to compare the performance in reckoning between peers from different countries, who have already been personally introduced to each other via internet)

• Opportunities for early language learning (for instance, when gathering virtual teams consisting of peers from different countries, and using the XO’s loudspeakers for generating words in the respective language).

• Support community building, local and international: The opportunities for international contacts as indicated above need to be complemented by local cooperation. Only local cooperation can optimize learning with respect to the local situations; this should be supported by Web 2.0 features:

  • A wiki describes the didactics of reckoning in primary education, while the description is open for personal amendments according to local specialties; a description generated and written in German (containing many pictures!) shall be available in the same wiki and might serve as a model.

  • Newsgroups shall offer opportunities to discuss didactics of evaluating primary education, and to exchange exercises.

  • Mailing lists serve for distribution of information (see the subsequent point).

• In this way European experiences can be transferred to developing countries in a bottom-up approach. The same might work for the following idea:

  • Develop a local didactical support system: An information structure built bottom-up as described above might be complemented by top-down approaches via UNESCO and ministries or UNI/IIST and universities. The installation of educational support systems might be even more efficient in developing countries.

The topic of the software developed first in this project has the potential for fruitful development assistance, provided appropriate contacts via internet at a full range from curriculum developers, educators, teachers to students. These contacts might start with predominant contributions from Europe, but seem to have good changes when shifted completely to local communities.

**CONCLUSION**

In conclusion, we see that the OLPC project itself has a great impact on bringing computer literacy to developing countries. In contrast, the role of
these mobile devices in the industrialized world (and in this particular example European world) rather differs. This is because there are comparatively less problems concerning hardware issues as well as software and internet availabilities and the research work had to concentrate on the use of digital devices on a very early stage of education. Bearing the increasing importance of digital literacy in mind, educational institutes must now consider how the integration of laptops can be done in a worthwhile fashion. As a result of this fact, the authors address to the question: Can education be improved by using such digital devices? This paper reported the first experiences done in Austria and stated that further studies will be necessary to enhance traditional education by using digital devices.

References

Prensky, M. (2006). Don’t bother me, Mom, I’m learning! How computer and video games are preparing your kids for 21st century success and how you can help! Paragon House, St. Paul, USA.


Resources


