Master Thesis

Sven Bergmann

Sugar – Not necessarily unhealthy

An examination of the XO used in primary schools

Master Thesis

Sugar – Not necessarily unhealthy

An examination of the XO used in primary schools

Presented to the Faculty of Mathematics and Computer Science
of the University of Bremen
in Partial Fulfilment
of the Requirements
for the Degree of

Master of Science in Digital Media

Sven Bergmann

Matriculation number 2174694
Grenzstrasse 3 - 21337 Lüneburg
iceberg@tzi.de

Approved by
Supervising Committee

Prof. Dr. Heidi Schelhowe
Prof. Dr. Jürgen Friedrich

Lüneburg, 2nd October 2009

Abstract

Since their foundation in 2005, the *One Laptop Per Child* project aims to equip children in developing nations with a low cost, robust and connected laptop, provided with a newly developed user interface called SUGAR, to support their education through creative and collaborative activities. Several schools around the world already received their laptops and reported primarily a positive feedback. According to this, the application of such an interface could also support children in industrial nations, where the *digital divide* can be high as well.
Subject of this work is the identification of the pros and cons of the SUGAR-interface in respect to usability, while handled by primary school children. But since SUGAR was especially developed for the OLPC project and is accordingly a relatively new interface, it will first be heuristically compared with an already established GUI to get an insight into its structure and features. Afterwards the results of an usability test, that was performed with seven children, will be presented and by means of defined usability criteria analysed.

Abstrakt

Seit ihrer Gründung im Jahre 2005, ist das Ziel des *One Laptop Per Child* Projektes, jedes in einem Entwicklungsland lebendem Kind mit einem günstigen, robusten und netzwerkfähigen Laptop auszustatten, welcher die neu entwickelte Benutzeroberfläche namens SUGAR beinhaltet, um ihre Kompetenzen mithilfe von kreativen Aktivitäten, die ein kooperatives Arbeiten ermöglichen, zu fördern. Größtenteils positive Rückmeldungen kamen von vielen Schulen auf der Welt, die bereits mit Computern ausgestattet wurden. Demnach könnte eine solche Oberfläche auch Kinder in Industrienationen unterstützen, denn dort ist die sogenannte *digitale Spaltung* ebenfalls zu beobachten.
Thema dieser Arbeit ist die Identifikation der Vor- und Nachteile der SUGAR-Oberfläche in Bezug auf Bedienbarkeit (usability), wenn diese von deutschen Grundschulkindern verwendet wird. Aufgrund der Tatsache, dass SUGAR speziell für das OLPC Projekt entwickelt wurde und demzufolge eine relativ neue Oberfläche ist, wird diese zunächst mit einem bereits etabliertem System heuristisch verglichen, um einen Einblick in Struktur und Funktionen zu erlangen. Anschließend werden die gesammelten Daten eines Usability-Tests, welcher mit sieben Kindern durchgeführt wurde, präsentiert und anhand von definierten Kriterien analysiert.

Contents

1 Introduction

"It is the most important thing [the One Laptop Per Child project; note from the author] I've ever done in my life [...] The reception it's received worldwide has been absolutely incredible. [...] The idea is simple - it's to look at education. This is an education project, not a laptop project. [...] If you take any world problem. Any issue on the planet, the big ones: peace, the environment, poverty. The solution to that problem certainly includes education; could even be just education. And if you have a solution that doesn't include education, it's not a real solution at all.
[...] our working premise or assumption is that if we can make education better, particularly primary and secondary schools, not tertiary education, the place would be a better world. And the second assumption just since this is a technology conference, is that in emerging nations – not emerging technologies – in emerging nations, the issue is not connectivity. [...] It *was* the issue; it's not a solved problem, but there are many people and many systems working on it. [...] It's happening; it doesn't need me, it doesn't MIT, it doesn't the Media Lab. But for education, the roadblock is the laptop."

Nicholas P. Negroponte [Negroponte, 2005]

The above quote by Nicholas P. Negroponte is from his presentation at the *Emerging Technologies Conference 2005* speaking about the *One Laptop Per Child* (OLPC) project. And the laptop he is refereeing to is the *Hundred Dollar Laptop*, also called XO or XO-1, which was developed by the OLPC project. It is a a robust, low-cost, low-power and connected laptop computer with a unique and novel graphical user interface (GUI). The projects' primary mission is to distribute these laptops to children in developing nations all over the world, to "create educational opportunities for the world's poorest children" [OLPC, 2009]. According to OLPC the laptop was deployed to more than 25 nations since the foundation of the project in January 2005, for instance Uruguay, Peru and Mexico, with positive feedback from children, teachers and parents. By now some initiatives were established in industrial nations, among others *OLPC Austria* and *OLPC Deutschland e. V*, who support and initiate new projects.

The GUI, which is called Sugar, was developed originally for the XO-1, but can meanwhile be installed on several other platforms too. It has a very compact program design and is based on the goal to "avoid bloated interfaces" and "limit the controls to those immediately relevant to the task at hand" [OLPC, 2009].

As mentioned, the laptop was originally developed for children in developing nations, but more and more initiatives are launched and XOs are also used in industrial nations, for example in Birmingham (USA), where according to Negroponte the *digital divide* is also very high and need to be resolved [Heise, 2007]. But why should such a laptop or just the Sugar-interface not be used in other schools as well? OLPC designed Sugar especially for children to support their education and get connected to each other. And if the system can help children in developing nations, maybe it is feasible and useful for all other children too. Since the *World Summit on the Information Society* (WSIS) 2003 in Geneva and 2005 in Turin, it is clear that social discrimination, as a result of the absence of competence when using digital media, may also occur in Europe [Wedekind and Koerber, 2009]. That could indicate that Sugar might be beneficial in Germany as well.

Due to the fact, that `Windows XP` is the most used operating system and GUI at present, – according to W3C [Network Solutions, 2009] – it can be assumed that this system is most likely also used in school. Not only for administrative work, but also in the students' computer room. This GUI is based on a more than thirty years old metaphor, the so-called *desktop metaphor*, which represents a real desk with all its elements and is still the standard. It was originally elaborated for office workers that are familiar with desk work, inboxes, files and folders. But is such a system appropriate for children, especially young children?
The possible application of the XO respectively its GUI Sugar in elementary schools is the background of this thesis. Correspondingly the laptop and the GUI will be

tested in school, more precise with students from a primary school in Germany. It should be mentioned that it would go beyond the scope of the thesis to examine if such a system is an appropriate tool for primary school children in general. Instead, after a comparison between `Windows XP` and SUGAR to detect the varieties between an *established* and a *new* system, SUGAR will be tested – by means of defined usability criteria and tasks – to identify its difficulties and vantages when primary school children handle the XO respectively SUGAR. Although the main focus is on SUGAR, thus on the software of the XO-1, it can not be avoided that partially some hardware aspects may also be considered. An example could be the computing power of the laptop.

The structure of the thesis is the following.

In the next chapter the OLPC initiative and their project goal are introduced. The XO is presented, by giving an overview about its hard- and software, especially the newly developed GUI called SUGAR. Results and experience of field reports and studies about the XO are pictured at the end of the chapter.

Chapter three is devoted to the domain of usability. The history of the term and the field of usability are explained, as well as a definition and approved norms that defines usability. In relation to the usability test with primary school children the sub-item of usability testing is additionally specified.

SUGAR is heuristically compared with another GUI in chapter four and that happens to be `Windows XP`, because it is the most successful and wide spread graphical user interface according to the statistics from W3Schools [Network Solutions, 2009]. The most clearly varieties between both systems, which are based on different theoretical approaches – `Windows XP` is based on the *desktop metaphor* and SUGAR on the *zoom metaphor* – will be examined.

The fifth chapter describes the SUGAR-tests that were performed with seven primary school children. Methodology, target group and procedure of the different meetings are described and the collected data is afterwards analysed and evaluated.

In the final chapter of the thesis the results are centralised, a conclusion is given and the essence is elaborated and set into the context. The report will be completed with a personal appraisal and outlook.

All data and statements of this thesis refer to the status as of publishing and the version of the SUGAR-interface that was used is build 767 (8.2.0).

2 OLPC

"Mission Statement: To create educational opportunities for the world's poorest children by providing each child with a rugged, low-cost, low-power, connected laptop with content and software designed for collaborative, joyful, self-empowered learning. When children have access to this type of tool they get engaged in their own education. They learn, share, create, and collaborate. They become connected to each other, to the world and to a brighter future."

OLPC [OLPC, 2009]

Everything started with a vision. In his book *Being Digital* [Negroponte, 1995] Nicholas Negroponte took the chance of taking a prospect on personal computing. He visioned "technologies that will make your telephone as context-sensitive as an English butler" and suggested what being digital could mean for our life in the future. With this book he described also the theory of *self-empowerment*. The core of this theory is fundamental to comprehend his vision: that education is a common method of resolution, according to the approach of helping others to help themselves. Out of this evolved the *One Laptop Per Child* project – its logo is presented in figure 2.1.

In this chapter the development of the OLPC project and the XO – the inexpensive laptop computer created by the OLPC – will be described, as well as an overview of existing studies and progress reports about the XO in action.

2.1 OLPC

The One Laptop Per Child Association, Inc. is a non-profit organisation located in Cambridge, Massachusetts and was founded at the World Economic Forum in Davos, Switzerland in January 2005 by faculty members of the Massachusetts Institute of Technology's (MIT) Media Lab. One of them was Nicholas Negroponte a Greek-American architect and computer-scientist. The MIT Media Lab was also set up by Negroponte and furthermore by Seymour Papert, a mathematician, computer scientist, and educator from South-Africa in 1985. Negroponte has ever since been the chairman OLPC. Seymour Papert, also a well known pioneer of artificial intelligence and inventor of the Logo programming language, is an adviser of the OLPC initiative. More information about Papert is given in 2.3.2 (A new interface).

Figure 2.1: OLPC Logo

The OLPC project wants to promote children's education in developing nations. Their primary mission is to make children "more active in their own learning, through collaborative and creative activities" [Negroponte, 2008]. To reach this goal OLPC wants to equip children around the globe no matter how poor or rich they are with their own laptop computer, which should only cost about $100. Therefore the OLPC project started to develop a laptop computer with a unique and novel interface from scratch. The first prototype of the initially called green machine was presented in November 2005 and is pictured in figure 2.2. Today the laptop is named *XO–1* and its GUI Sugar. Both are described in more detail in 2.2 (XO-1).

Commonly the laptops are sold to governments and distributed by the countries educational ministries. However, anybody can donate one or more laptops via OLPCs' website for developing nations. Initially no consumer version of the XO was planned, but from November 12, 2007 until December 31, 2007 some 83,500 laptops were ordered within the *Give One Get One* (G1G1) program, where two XOs are bought, one is sent to the purchaser and the other one is donated. A second G1G1 program ran between November 17, 2008 and December 31, 2008 through `Amazon.com` and `Amazon.co.uk` but only 12,500 pieces were sold.

Since the beginning of the project the OLPC initiative has stuck to their vision that the laptop should only cost $100. A great number of units realised through mass production that started in 2007 was planned, in order to facilitate a low price of the computer. OLPC has not reached this target so far due to bad sales figures. The actual production costs are around $150 to $175 per unit and probably it will take some more years to reach the aspired price of 100 dollar [pressetext, 2006].

As already mentioned above *One Laptop Per Child* is not only the name of the company and the project, but also their vision. They want to create educational opportunities for the world poorest children by providing each and everyone with a robust, low-cost, low-power and connected laptop. Five core principals [OLPC, 2009] define the goal of the initiative. These five principles are:

Child ownership: OLPC developed the XO explicitly for children in primary school and they believe that it is a fundamental right of each child to own a personal computer. This results in new tasks and responsibilities for the children: to take care of the laptop, protect and share it with others. The laptop can take the school out of the school, it is a mobile learning and teaching tool. So the children have the possibility to learn and discover new things, wherever and whenever they want.

Low ages: As already mentioned, the XO is designed for children in primary school, especially between 6 and 12. Therefore it is important that handling the laptop is feasible without being able to read or write, just by playing around with it. For this reason symbols are as often used as possible instead of written text. OLPC emphasised that we all should know that playing is the basis of learning and helps to achieve skills like reading and writing.

Saturation: OLPC wants to gain a high level of saturation in a given environment. That means only whole classrooms, villages or even cities should be equipped with laptops wherever applicable.

Connection: The connectivity of the XO has so far been unique (cf. 2.2 (XO-1)). If one laptop is connected to the internet, this connection is shared with all other XOs in range. And even if one of them is in idle state, it still provides internet access to other XOs. Additionally this connectivity enables the children to work together on different activities provided by the laptop, like painting, composing music or even programming.

Free and open source: The last core principal of the OLPC project is that the whole software that is running on the XO is free and open source (FOSS). The main benefit is that the software of the XO can be upgraded or newly developed by the huge open source community and any other person. This also allows everyone (private persons, schools, department of education, etc.) to adapt the software according to their own requirements.

To sum up, the aim of the OLPC project is to bring education to children all over the world, by means of a laptop computer that connects them to each other and to the entire world.

The OLPC project had to deal also with difficulties and critique during their work. Their biggest disappointment is, as explained above, the non-achievement of the aimed price of the XO. OLPC was not able to sell as many laptops as needed to produce the XO to the price of $199. According to OLPC this is caused by the politicians in developing nations, who are not willed to engage themselves in such a project, because it is to risky to invest in such a project in their opinion [Fildes, 2007]. Another recoil for instance was in November 2007, when OLPC was denounced for the unauthorised usage of a patented keyboard-technique of a Nigerian company.

The major critique about the project was *laptops are not food* and that children in third world countries need food and medicine instead of laptops. OLPC replied that besides food, water, medicine and shelter, education is also a very important cause, and for that OLPC "is trying to get this 'simple gadget' (a full-blown laptop) into the hands of kids for them to learn, communicate, interact, and many other things..." [OLPC, 2009].

The Microsoft founder Bill Gates considered the spread of the XO as unimportant [Fried and Beiersmann, 2008] and proposed cellphones as an alternative [Perton, 2006]. During this year Negroponte himself criticised SUGAR, but only as an operating system. He said that it "should have been an application on a normal operating system" [Jackson, 2009] without dealing for example with power and wireless management. In this context has to be mentioned that since over one year Microsoft is working on a `Windows XP` version for the XO. The laptop has an *dual-boot capability* and is able to boot into Linux (including SUGAR) and `Windows XP`. It is planned to sell the laptop with `Windows XP` sometime in 2009.

In the next section the laptop of the OLPC initiative and the newly developed GUI will be introduced.

2.2 XO-1

The XO–1 or XO that is also called \$100 Laptop, as displayed in figure 2.2, is an educational tool developed by the OLPC project. It is designed for children all around the world, but primarily for children in developing countries. It is constructed to run under tough conditions and in remote areas, which means it is robust (resistant to dirt and moisture), energy efficient (needs only 2 W of power during normal usage) and costs much less than other laptop computers (OLPCs' original goal was to reach \$100 – production costs are today at around \$150 to \$175).

Figure 2.2: The XO-1

2.2.1 Hardware

Within this subsection the basic hardware of the XO is presented.

Core elements

The CPU (central processing unit) is a x86-compatible processor with a clock speed of 433 Mhz. The graphics controller is integrated into the CPU and the DRAM memory is 256 MiB. The XO has a mass storage of 1024 MiB SLC NAND flash, but no hard disk or optical drive. Three USB ports and a SD card slot are built in. Furthermore the XO has a sealed rubber-membrane key-switch keyboard, a gamepad, an extra-wide touchpad, a built-in camera and connectors for a headphone and a microphone. The microphone input is also operable for other input-signals, for instance from a microscope.

Figure 2.3: eBook Mode

The XO is obviously not the fastest computer on the market, but according to OLPC its power is sufficient enough that Sugar and its activities run with an acceptable speed.

Display

The display is one basic element of the XO related to energy saving. The display can be used in darkness and also in bright sunlight, "because learning takes place in both" [OLPC, 2009]. The screen is a *combination* of two screens: a backlit screen and a reflective screen. When working in direct sunlight the backlight doesn't matter and can be turned off, because the reflective monochrome screen reflects the light; no colours are shown and low energy is used. When working in complete darkness – no reflection on the screen – the backlight needs to be active. In-between a mixture of both screens makes the display readable. The display is rotatable and can be turned upside down on the keyboard (cf. figure 2.3) similar to a Tablet PC.

Keyboard

The whole keyboard area of the XO is covered with rubber to protect the device. The layout is more or less comparable to common keyboards embedded in laptop computers. In addition there are some special keys for the unique functions of the SUGAR-interface, e.g. a search key, a button for each of the four views (cf. 2.3.3 (Zoom Levels)).

Touchpad

The XO has a built-in track-pad, more precisely a track-pad with three areas. Therefore it is wider than normal pads, but unfortunately only the middle area has capacitance and responds to finger contact. The original idea was, that all three areas should serve with a stylus as a graphic tablet. But until know no stylus or software had been announced. Two buttons are positioned under the track-pad. The left button is for element selection and with the right button contextual menus can be activated.

Battery Life

As already explained the XO needs only 2 W of power during normal usage – conventional laptops need 10 W to 45 W. OLPC is working on a low power e-book mode, when the display is turned around like in figure 2.3 and text is shown only via the monochrome display. The power consumption is estimated to be 0.3 W to 0.8 W which would extend the battery life enormously.

The battery can be charged in several ways. First of all by using the AC adapter for a power network and if there is none available some alternative charging methods are possible: e.g. a hand-crank or a foot-pump.

Network

Because the laptop will be deployed in areas with very little or no infrastructure and the OLPC project wants to help kids get connected to the world, they concentrate on a wireless solution for sharing thoughts, ideas and activities via the network. The XO is equipped with two adjustable, rotating antennas (cf. figure 2.2) that support diversity reception for wireless connectivity.

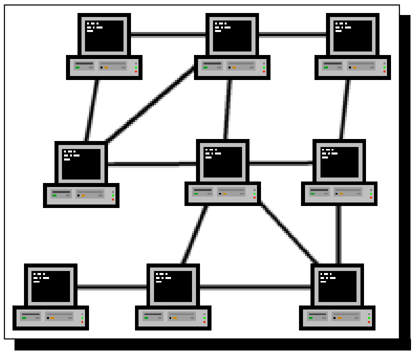

Figure 2.4: Mesh network

When using the laptop it connects automatically to any reachable mobile ad-hoc network. A mobile ad-hoc network – comparable to a peer-to-peer network – exists of two or more computers that connect autonomously to a network, where every computer is connected to all others (directly or indirect) and form a so-called mesh-network cloud, as displayed in figure 2.4. If one of the laptops within the cloud has access to the internet all other computers are able to use and share that connection. And even if the XO is in idle state it is still capable of mesh operations, meaning it is still part of the network and can forward data packages across the cloud. Another advantage using mesh network technology is the capability of self-healing, if one laptop drops out, the other XOs are able to rearrange and try to maintain the connection for all laptops in the mesh.

An established connection to other XOs enables to communicate and share information. Users can chat, write a text-document or paint a picture together. Each program that is installed on the laptop – it has to be installed on all participating XOs – and each object that is created can be shared with others.

The two antennas are more powerful than conventional wireless antennas and able to receive and send data over some hundred meters without package loss. A test in the Australian Outback revealed that it is possible to send data over a distance of 1260m (line of sight) with 1Mb/s and some ping losses [Vota, 2007]. At a distance of around 600m all packages were sent with 1Mb/s without any problems. Nevertheless the data rate across a wide spread mesh network is not high, but sufficient enough for normal communication like browsing and mail.

2.2.2 Software

The XO is shipped with a FOSS software package based on the Fedora Linux distribution including the newly developed interface called SUGAR and several programs, the so-called activities. Both the user interface and its new structure are described in more detail in the next section, but some basic features should be mentioned here.

The first one concerns the software costs. SUGAR and all its components are FOSS, which provides the opportunity that any developer can participate and contribute new ideas, improvements and applications. So it is guaranteed that there will be no additional costs for software when using the XO in its original purpose.

The second feature deals with the developers' aim to write a software that is readable and feasible for children all around the world. The OLPC project points out in their human interface guidelines that

> "Developers must keep in mind the broad range of cultures and languages that the laptops must transcend. In particular, activities should not depend on western icons and modes of thinking, but should abstract ideas to a level that would be familiar to humankind in general, where possible." [OLPC, 2009]

The usage of written language within SUGAR is as minimal as possible, so that children who are not able to read can nevertheless use the XO. Applications are launched by clicking the associated icon like shown in figure 2.5. The icon on the top left represents the Journal activity (cf. 2.3.5 (The Journal)), the speech bubble is the chatting icon, the earth is connected to the Browser activity and the mixed pallet with the brush represents the painting program.

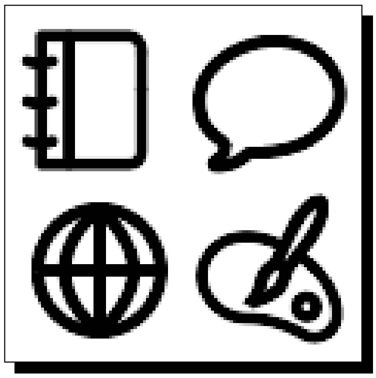

Figure 2.5: Activity Icons

Even though SUGAR was developed especially for the XO, it can practically run on any existing platform today, whether installed or booted from a live CD. SUGAR and its history of origins are described in more detail in the next section.

2.3 Sugar

In contrast to other operating systems, SUGAR is relatively unknown. Therefore it is necessary to introduce this GUI before comparing it to `Windows XP` in the fourth chapter of the thesis. After describing both the desktop and the zoom metaphor in the following, the layout and features of SUGAR are presented.

2.3.1 Desktop Story

Most of the common used GUIs today, like Windows, Mac OS and the X-Window System, are based on the *desktop metaphor* that was developed initially for the *Alto computer* more than 30 years ago at the Xerox Palo Alto Research Center (PARC). At the PARC the first laser printer, the ethernet, the first concept of a personal computer and its GUI were developed. The desktop metaphor again is based on the so-called interface metaphor, a class of user interface procedures, actions and visuals which utilise certain cognitions that users already have of other areas and help them to interact more easily with a computer. Correspondingly the desktop metaphor defines the GUI of an operating system as an image of a real desk, whereon documents and folders of documents can be placed. These documents can be opened in their own window, representing a real paper copy lying on the desk.

The evolution of this desktop-concept was based on the aspired market for the *Alto computer* and the developers at PARC had an exact idea of their customers in mind: the 500 largest companies according to the *Fortune 500* list of large companies [Cable News Network, 2009]. Not only because of the price – the parts for the *Alto* were about $16.000 at that time –, but also because they were the only ones who could use a computer to support their operational procedures. There was no need for home users or even children to have a computer. Thornburg declares that “In the highly document-centered world of business, it only made sense for XEROX (the ’document’ company) to provide an interface that was comfortable for corporate executives” [Thornburg, 2008, p.4]. And so an image of the business worlds’ work environment emerged as a computer interface with documents, folders, in- and out-boxes (for retrieving and sending messages) and a clipboard. Furthermore some tools are available, like a calculator, a notepad or a paint utility.

Until today this metaphor is the basis of commonly used user interfaces. During the evolution of these interfaces the user was more and more emphasised in the development process, in order to improve its level of usability. In this context an apt quotation about user interface design from Alan Kay found in *Das Medium aus der Maschine* from Heidi Schelhowe [Schelhowe, 1997, p.163] should be mentioned. Kay is an American computer scientists, known for early object-oriented programming and designing of user interfaces and also a advisor of the OLPC project. He said

> “[...] that the actual dawn of user interface design first happened when computer designers finally notices, not just that end users had functioning minds, but that a better understanding of how those minds worked would completely shift the paradigm of interaction.” [Kay, 1990, p.192]

Schelhowe added that the interest how the human mind is functioning, is in conjunction with the interest to design an interface in a way that human behaviour and computer operations are as engaged as possible and that the user is eventually in the main focus of the development of personal computers and their user interface [Schelhowe, 1997, p.163].

Accordingly the desktop metaphor was also the basis of the user interface of the *Apple II* in 1977, the worlds first *personal* computer. The Apple II was less expensive than the Alto and thus designed for a wider range of people.

2.3.2 A new interface

Because of the revolutionary vision of the OLPC initiative to equip every child with a laptop and of its interesting and playful design, the XO is since its release in focus of the press and media. Unfortunately the much more interesting part of the product – the XO is still a new looking laptop, but constructed with already available hardware – has taken a back seat: the Phython-based GUI of the XO, called SUGAR. It is the "first truly new developed user interface for personal computers" [Thornburg, 2008, p.2] since the mid-1970's, when the first GUI based on the desktop metaphor was developed at the Xerox Palo Alto Research Center (PARC) as explained in 2.3.1 (Desktop Story).

The reason why no new interface was designed since three decades is justified by David Thornburg, an American author, consultant and formal employee of the Xerox Palo Alto Research Center (PARC), as follows:

> "[...] either we [the employees of the Xerox PARC; note from the author] were the brightest designers on the planet (unlikely), or that users didn't pressure the industry to revisit the topic of user interface design (more likely)." [Thornburg, 2008, p.3]

The development of the SUGAR-interface was influenced by Seymour Papert, who founded the MIT Media Lab with Nicholas Negroponte as already explained. When he talked about "children using computers as instruments for learning and for enhancing creativity" [MaMaMedia Inc., 2009] in the sixties, he was not taken serious, but it was foremost in his lab where children had the chance to sit in front a computer and use it for writing and drawing.

Through his work with Jean Piaget, a Swiss philosopher, natural scientist and developmental theorist who is "the pioneer of the constructivist theory – constructivism is a psychological theory of knowledge and defines that human experience and learning is based on constructional processes – of knowing" [von Glasersfeld, 1990, p.4], Papert went beyond "constructivism to constructionism"[Thornburg, 2008, p.5], where the learner is an artefact creator and shares these artefacts with others. His position is that this should also apply to children, who should *do* something and learn on their own – particularly with computers – instead of teaching them. According to Papert computers are "tools to think with" [Papert, 1981].

Paperts definition of constructionism is the following:

> "The word constructionism is a mnemonic for two aspects of the theory of science education underlying this project. From constructivist theories of psychology we take a view of learning as a reconstruction rather than as a transmission of knowledge. Then we extend the idea of manipulative

> materials to the idea that learning is most effective when part of an activity the learner experiences as constructing a meaningful product." [Papert, 1991]

In the book *Constructionism in Practice* Kafai and Resnick point out that:

> "Children don't get ideas; they make ideas. Moreover, constructionism suggests that learners are particularly likely to make new ideas when they are actively engaged in making some type of external artifact – be it a robot, a poem, a sand castle, or a computer program – which they can reflect upon and share with others. Thus, constructionism involves two intertwined types of construction: the construction of knowledge in the context of building personally meaningful artifacts." [Kafai and Resnick, 1996]

OLPC expresses in their human interface guidelines [OLPC, 2009], that SUGAR provides the opportunity to build meaningful artefacts by having activities (cf. 2.3.6 (Activities)) instead of applications. This "[...] represents an intrinsic quality of the learning experience we [the OLPC, note from the author] hope the children will have when using the laptop" [OLPC, 2009]. Furthermore they explain that this "approach focuses on thinking, expressing and communicating with technology" and they "hope to make the primary activity of the children one of creative expression, in whatever form that might take"[OLPC, 2009]. Most activities of SUGAR focuses on the creation of objects, for example painting a picture, composing a song or writing a program. When the XO is connected to other XOs it is possible to create objects with other users.

An overview about the layout and features of the SUGAR-interface is given in the following subsections.

2.3.3 Zoom Levels

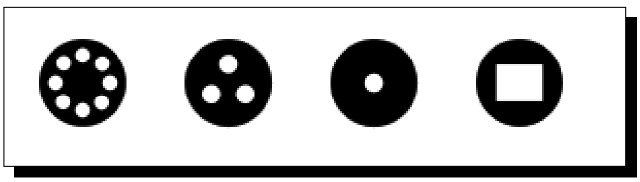

Figure 2.6: SUGAR zoom buttons

As already mentioned, mesh networking is one feature of the XO. This ability is explicitly represented in the GUI of the laptop: with the possibility of changing its *view*. The so-called *zoom metaphor* is a new approach of constructing an interface unlike the desktop metaphor that is is common for GUIs nowadays. It describes four distinct zoom levels of SUGAR that shows different layers of the GUI. These four levels, as described in the OLPC human interface guidelines [OLPC, 2009], are the *neighbourhood view* that shows the entire network environment, the *groups view* with invited friends that are connected, the *home view* from where all activities can be launched and the *activity view* that shows the current active activity. Switching between these views is possible with the zoom buttons on the keyboard displayed in figure 2.6 or the icons within the Sugar-Frame (cf. 2.3.4 (The Frame)).

An outline of the layers is shown in figure 2.7 and they are described in more detail in the following.

The Home Screen

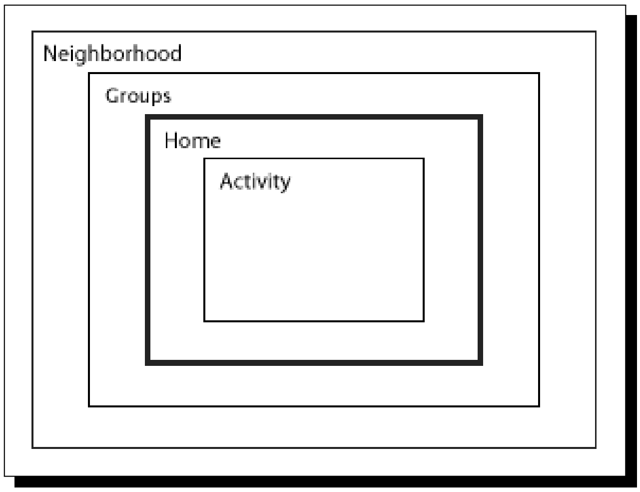

Figure 2.7: Sugar zoom levels

The welcome- or home screen of Sugar is shown in figure 2.8. This is the only view that looks as similar as possible to a desktop based on the desktop metaphor explained in 2.3 (Sugar). At the centre of the screen is the XO icon that represents the user. Each child can select a stroke and fill colour for their XO out of 18 colours (6 colours in 3 hues). According to that 306 different XO icons are possible. The ring around the XO, called the Sugar-ring, shows all favourite activities. Activities can be marked as a favourite within the list view of all activities. This list can be activated by clicking on the list view icon in the top right corner, while being on the home screen. The icon beneath the user icon shows the current active activity, which can only be one at a time. If the user wants to switch between loaded activities he or she has to click on the appropriate icon in the upper part of the Frame, which is describe in 2.3.4 (The Frame). With a click on an icon a context menu appears with additional options. This applies not only for the activity icons in the Sugar-ring, but also for all icons that are part of the Frame and the user icon in the centre. The context menu of the user icon offers the possibility of launching the control panel, restarting or shutting down the system.

Neighbourhood View

The view without any zoom shows the mesh network or neighbourhood view (cf. figure 2.9). The neighbourhood view shows all users and access points in the environment. From here other users can be invited to be a friend, member of a group or participate in an activity.

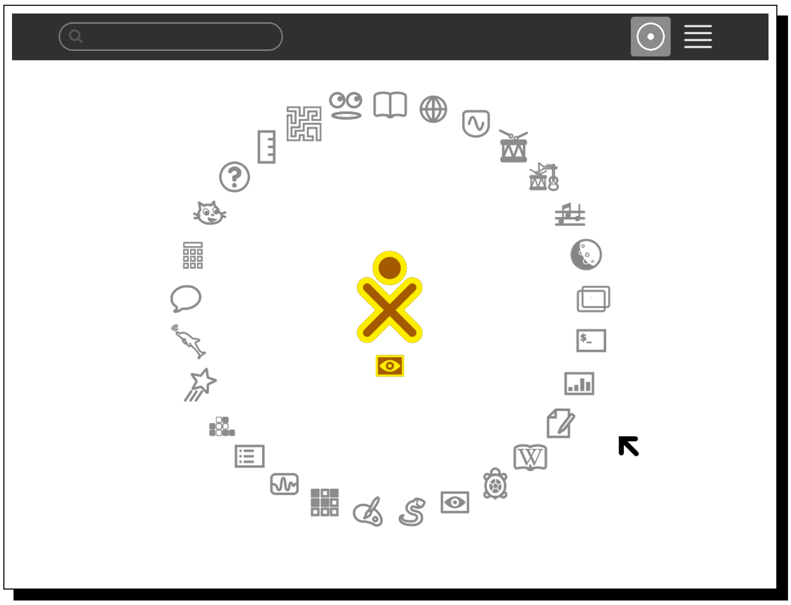

Figure 2.8: Sugar Home screen

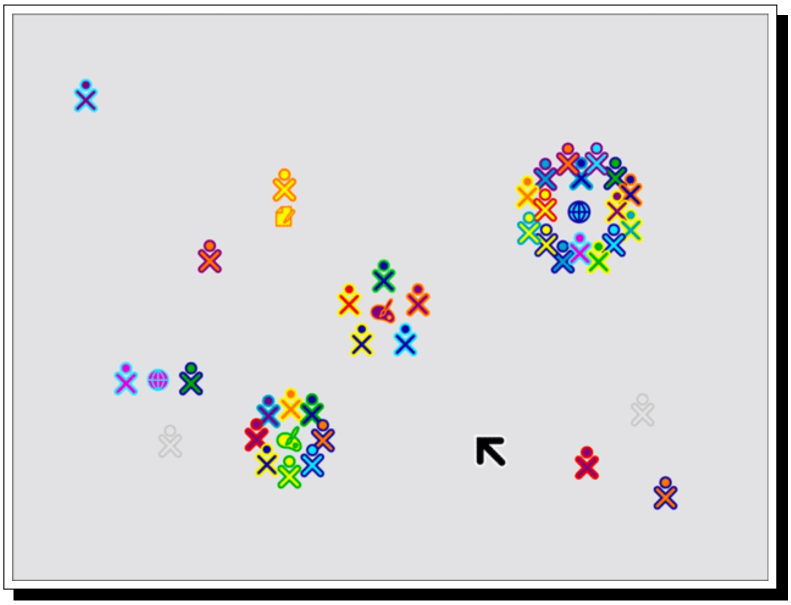

Figure 2.9: Sugar neighbourhood view

Groups View

In figure 2.10 the group view of the interface is shown. This view is located between the home and the neighbourhood view and shows all friends and groups of the user. In this view it is possible to create groups and add friends to a specific group. An individual arrangement of the users friends' is possible, e.g. by name or icon colour. All activities a child is performing can be used with others – assumed that the activity is installed on all participating systems – through the group view and create objects like a story or a picture together. If a friend or a whole group work together on an activity respectively on an object, the icon of the members are arranged around the activity symbol.

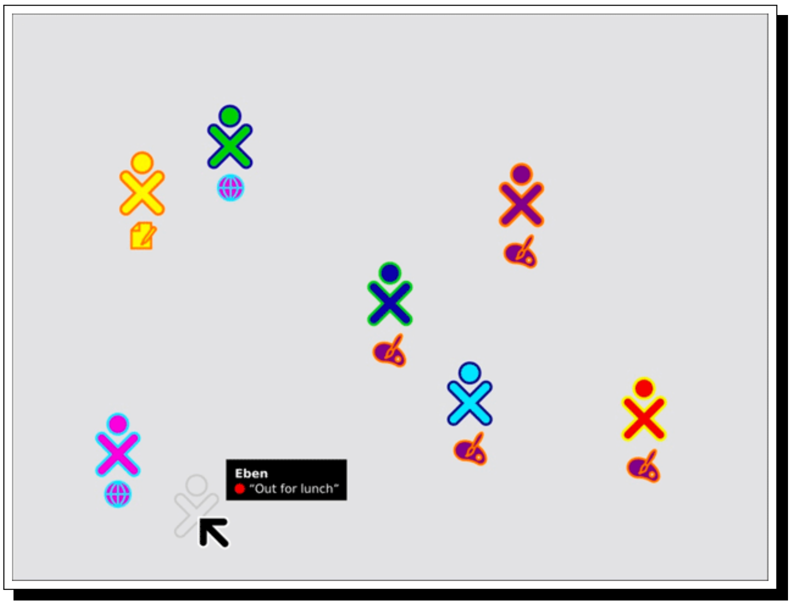

Figure 2.10: SUGAR group view

Activity View

The area with the greatest zoom, is the activity view. One example of an activity, the Browser activity, is displayed in figure 2.11. Activities are the equivalent to programs that run under other operating systems (cf. 2.3.6 (Activities)). Even if more activities are launched, only one activity can be the main active activity. It is not possible to arrange two or more activities on the screen at the same time, as for instance in `Windows XP`. Switching between activities is possible at any time via the corresponding icon in the SUGAR-*Frame* (cf. 2.3.4 (The Frame)). An activity can be closed by clicking on the close symbol or by clicking on the close dialogue that appears while hovering on the activity icon. The Frame is described in the next subsection.

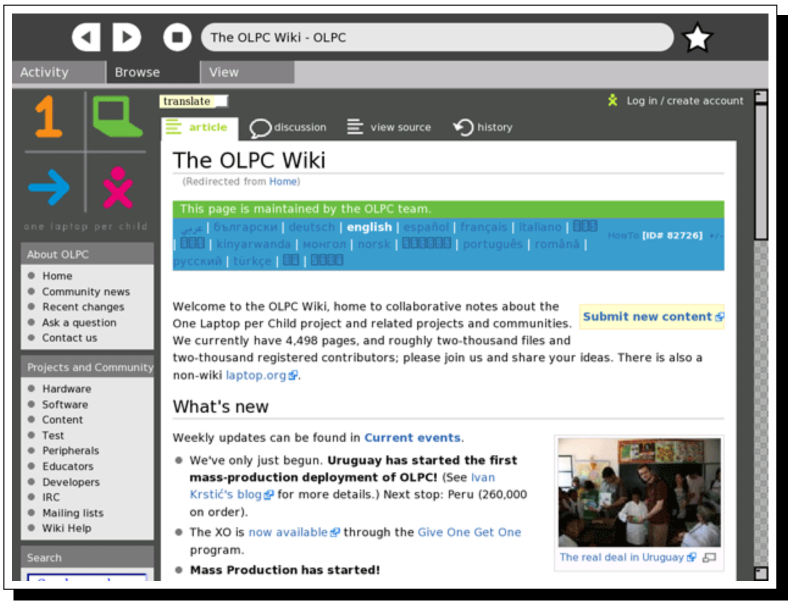

Figure 2.11: SUGAR Browser activity

2.3.4 The Frame

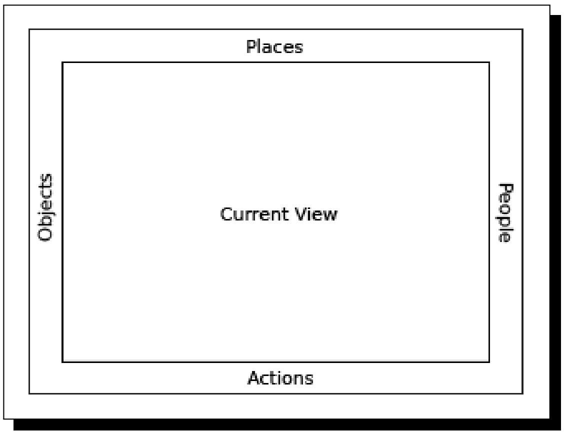

Figure 2.12: SUGAR Frame (layout)

In order to switch for instance between activities and the different views, SUGAR contains a form of menu bar, the so-called *Frame*. The Frame – see figure 2.12 for a sketch and figure 2.13 for a screenshot – consists of four areas: *objects*, *places*, *people* and *actions*. During normal usage of the XO this Frame is hidden, but can be faded in any time by moving the cursor to one of the four corners or edges of the screen (adjustable in the settings) or via the associated function key in the upper right corner of the keyboard. The Frame is automatically brought into view by any urgent system notification, for example low battery level. Its four areas are described in more detail in the following.

Objects

The left region of the Frame is reserved for temporary storage of objects. It is comparable to the clipboard in other GUIs. Via keyboard shortcuts or drag and drop any file or selection can be copied to the object area of the Frame. The objects are arranged as in a push-down stack, which means that the latest added object is

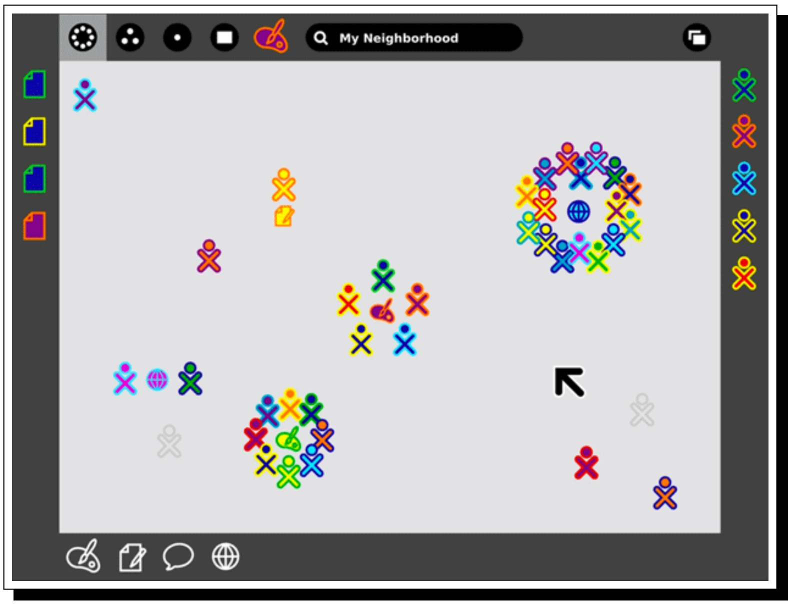

Figure 2.13: SUGAR Frame

on top. This feature allows the user to copy data from one activity and use it in any other activity, for example to embed a recorded photo in a text document.

Places

The upper part of the Frame provides the *places*-area of SUGAR. Here it is possible to switch between the views as described in 2.3.3 (Zoom Levels) with the four so-called zoom buttons. The grey circle on the left with a dotted-ring (activated in figure 2.13) represents the neighbourhood view. To the left there is the icon with three white dots that is associated with the group view. The one white dot represents the home view and the white rectangle links to the activity view. These buttons are also part of the keyboard layout as mentioned in 2.2.1 (Keyboard). Additionally the icons of all launched activities appear next to the zoom buttons. So it is possible to switch between launched activities without returning to the home screen.

Not yet integrated, but by now worth mentioning is the bulletin board that is planned to be part of the *places*-area respectively a button to the bulletin board. Within the bulletin board it should be possible to post objects, like texts, graphics or even music. It is like an advanced clipboard, where each element can be placed as on an information panel. Every view has its own bulletin board, so that each board is only visible to its respective users. The bulletin board enables communication and sharing between the users. For example users can discuss things via the board through chat bubbles or they share their created objects.

People

All individuals of the ongoing activity a user is collaborating with are shown in the right region of the Frame. His or her own icon is always on top. So a user is permanently aware of the users she or he is collaborating with. By hovering over a users icon biographical information – if available – is shown: name, age, hobbies and other personal information.

Actions

The bottom of the Frame shows incoming invitations, for example to join a drawing activity and system notifications, like a file download. Additionally the status of network, battery and audio volume are displayed.

2.3.5 The Journal

The XO is not equipped with a common file manager like the *Windows Explorer* in *Microsoft Windows* or the *Finder* in *MacOS*. Within the Sugar GUI another way of file management is used. Based on the widespread activity of keeping a diary, like some children also do, Sugar automatically keeps record of all activities a user is performing, e.g. if a text is typed or a picture drawn, the system stores these objects automatically – in the Journal. All entries can be seen and searched through to find any created object and in addition the user can also add personal data.

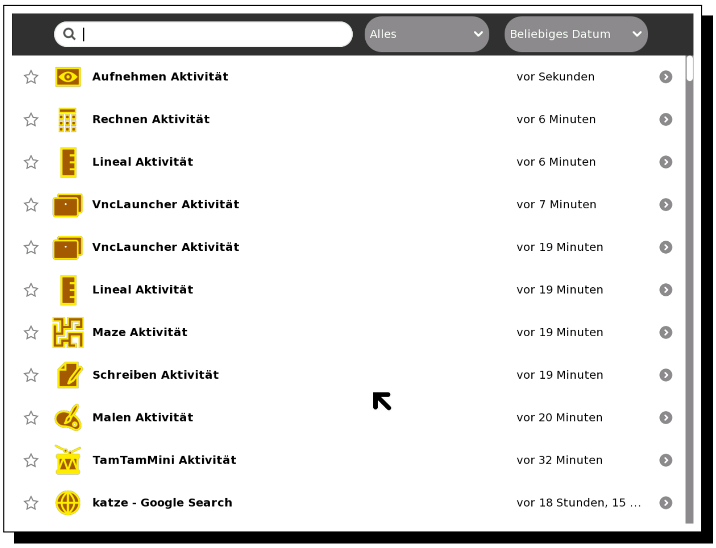

Figure 2.14: Sugar Journal

There are different types of Journal entries:

Implicit: Entries that represent the interaction with the computer e.g. activities, taking a photo or finished file download.

Note: Self created entries, comparable to a common Journal entry – these entries can be turned into a *to-do* note with an additional checkbox. The Journal filter function allows the user to see only the to-do entries.

Clipping: Any kind of object, like a text, a web-link or a photo, can be stored in the Journal for further usage.

Event: Events are entries that will take place in the future e.g. every accepted invitation to an activity that is not immediate but in the future, creates an event-entry in the Journal, so it can serve as a planning system. An audio alarm can also be added to these entries.

Progress Indicator: Entries with no immediate task-results e.g. progress bar during a file download.

The reason why the OLPC initiative has chosen such a way of file management, is because they "believe that the traditional 'open' and 'save' model commonly used for files today will fade away" and "the children who use them [the XOs; note from the author] will probably never see one of these obsolete devices (floppy disk icons)" [OLPC, 2009]. Instead of using these obsolete devices that we use today to save our files on any kind of storage medium, they think a more general notion of *keeping* things on a computer will become accepted and the idea of the Journal "as a time- based view of a child's activities" [OLPC, 2009] is their alternative concept of keeping things. For example "to 'open' a drawing you've kept, you simply resume it."[OLPC, 2009]

2.3.6 Activities

As already explained, the equivalent to common programs under Windows, Mac OS and other operating systems are called *activities* in SUGAR. They enable and support the user to perform different tasks. For example the Browser activity allows to visit websites on the internet, pictures can be created with the drawing activity and music can be produced with the *Tam Tam* activities. A special feature of SUGAR is that these activities are always displayed fullscreen, therefore the user can concentrate on one task completely. Every activity that is installed on all participating systems can be used with other XO users in the network.

2.4 XO-Experience

Since the foundation of the OLPC project in 2005 and the fabrication of the first XOs via production line in November 2006, a huge number of XOs has been manufactured. According to the last released deployment information from 4th January 2009, about 564.000 number of XOs have been shipped to several countries all over the world, for instance to Uruguay (202.000), Peru (145.000), Mexico (50.000) and Birmingham in the USA (14.000) [OLPC, 2009]. In course of the *Give One Get One* (G1G1) campaign 72.000 units were delivered. To get an insight into the success, the influence and the importance of the OLPC project, this subchapter contains some information from field reports and studies about the usage of the XO around the globe.

2.4.1 Field reports

Generally speaking all reviewed field reports present a very positive feedback on the usage of the XO. Teachers as well as the children enjoyed working with the XO. For example, 150 students from first to sixth grade of a school in Villa Cardal in Uruguay received their own laptop in May 2007. Not only for the children, but also for the teachers, it was their first contact to a computer as well as access to the internet. The teachers were trained, therefore they could show their students how to work with the XO. Hourcade et al. asked the fourth grade students what they liked and what they disliked about the XO. Most of the children liked the browsing and drawing activity, the possibility to take photos and record videos. More precisely Hourcade et al. reported that the children "enjoyed looking for information, sending mail messages, playing games, and uploading pictures and texts to the school blog" [Hourcade et al., 2008a, p.4]. They also noticed, that the children were not afraid of using an activity when the user interface was only available in English. On the other hand they were unsatisfied with the connectivity due to network problems at school and home, and the Journal, which the children found difficult to understand. Some students also had problems with pointing and clicking, because they were not used to working with a touchpad. The teachers in the school of Villa Cardal reported that the pupils were more motivated to read, write and share their documents with others. They were also pleased with the opportunity of having access to a wider set of material.

In June 2008 OLPC trials were started in Gaire and Dreikikir, Papua New Guinea, with 3rd graders of each school. David Leeming a British technical advisor visited the schools in November 2008 and got comparable impressions [Leeming, 2008] as Hourcade et al. in Uruguay. The teachers were extremely enthusiastic and very impressed by the increased motivation and creativity of some students ("standard of drawings using paint is quite out of the league of the adults!" [Leeming, 2008, p.4]). They also noticed an improvement in terms of engagement in learning and were amazed about the students' learning capacity. The most conspicuous negative observation was, that some students from other grades not possessing an XO were demoralised.

Another example is *OLPC Afghanistan* [Derndorfer et al., 2009], which was launched in November 2008. So far 450 laptops have been passed out to children of Iesteqlal High School in Jalalalbad. The goal is still to equip every single child in grades one to six (1.020 children in total), which should be accomplished sometime in 2009. Svetlana Senajova, one of the volunteers on-site is extremely confident about the success of the project. Via the SUGAR Labs mailing list in October 2008 she said:

> "Afghanistan could be the case study that shows the ability of the XO to be at the center of redevelopment. If we can make the XO work here - then we can prove to the world that it can work almost anywhere!" [Senajova, 2008]

2.4.2 Studies

Lowes and Luhr [Lowes and Luhr, 2008] published in June 2008 their evaluation results of a pilot project called *Teaching Matters* with sixth-grade students at the middle school KAPPA IV in Harlem, New York. Teaching Matters is a non-profit professional development organisation that partners with educators to improve public schools. Based on surveys with the students and their parents, they examined how students used the XO, both in school and at home. To get an insight into their findings some of the results are shown in the following.

What the students ...

liked: the typing (it is much faster than handwriting); the internet research; the camera; many of the programs; the portability and the look of the XO.

disliked: the XO is too slow, it frequently freezes and has an inaccurate cursor.

The most used activities related to schoolwork that were used by nearly all 21 students were the Browser, (21 pupils), the Write (20) and Chat activity (20). Whether they were using the XO for writing, chatting or else, "they were in every case sharing their thoughts or sharing their work" [Lowes and Luhr, 2008, p.7]. The students were eventually allowed to take the XO home, where other activities were used compared to school: taking photos (21), the Journal activity (20), the Browser (19) and record videos (19) or audio files (18). Teachers and parents had the impression the students were learning more while working with the XO and some even worked harder with the XO. Lowes and Luhr concluded the pilot project as a success for all participants: students, teachers and parents.

Figure 2.15: XO Rack

In a formative evaluation Uttam Sharma, Ph.D. student at the University of Minnesota, investigated the usage of the XO at two schools in Lalitpur, a district of Nepal [Sharma, 2008]. 31 out of 135 students of grade two and grade six of both schools who tested the laptop in school and at home were interviewed. In addition, 17 teachers were also consulted about their impressions. Just as in the already mentioned reports and study above, the pros outweigh the cons. All 17 teachers had the impression, that the use of computers even helped their teaching. It made the classes more interactive and the students were more interested in their studies. They "find the laptop and its layout easy to use. They think it is very intuitive and also feel that the students should not have much difficulty" [Sharma, 2008]. 95% of the students responded that using the XO made learning easier. The XOs' activities were very enjoyable for most students. They liked the possibility of doing activities as often as they wanted to and at their own pace. Nevertheless negative aspects were mentioned as well, of which the biggest problem was the jumpy cursor half of the students had difficulties controlling it.

A custom-built rack was manufactured to charge the XOs in Lalitpur, as displayed in figure 2.15.

2.4.3 Interim balance

In contrast to the recurrent criticism about the usage of computers in schools which act on the assumption, that computers isolate children and gets in their way of collaboration and communication [Alliance for Childhood, 2001], the observations and findings of the reviewed reports and studies showed exactly the opposite. For example, according to Hourcade et al. the XOs "encouraged and facilitated social interactions" [Hourcade et al., 2008b, p.1]. The children showed each other their creations and what they found while playing around, like games or information from the internet. They were interested what others were doing with the laptop and also helped each other. "These social behaviours made it so that knowledge about how to do something or how to access content would quickly spread throughout the classroom" [Hourcade et al., 2008a, p.6]. Another interesting point is, that some children took the role of teachers at home and showed their parents and siblings how to use the XO.

Concerning the central question of the thesis – pros and cons of SUGAR (cf. 1 (Introduction)) – the results of the test with primary school children will show if the above-mentioned corresponding observations of the studies and reports also apply:

- SUGAR is intuitive
- Children were not afraid of non-translated text
- XO is too slow
- XO freezes frequently
- Cursor is inaccurate and difficult to use

In this context it can be assumed that most of the primary school children in Germany have already worked with computers – according to the results of the *Kim-*

Studie 2008 (cf. 5.2 (Target Group and Setup)). And maybe their experiences are that advanced that they treat the XO more as a toy instead of a learning tool, for instance because of the *small* display and the low processing power. Another important question is, to which extend is SUGAR able to prepare children for their later working environment, where the most computers are equipped with other operating systems, mostly `Windows XP` [Network Solutions, 2009]. If answering these questions is possible with the results of the SUGAR-test with primary school children remains to be seen (cf. 6 (Conclusion)).

Before SUGAR is being compared with `Windows XP` and tested with primary school children, the field of usability will be described in the next chapter.

3 Being usable

"Usability is like oxygen – you never notice it until it is missing..."

Unknown author [Bigham, 2007, p.8]

Usability plays an essential role in the developing or upgrading process of any user-operated system respectively its GUI. User-operated systems are systems that a user is able to interact with via a GUI, for example a website, a software application or a mobile technology. Usability is an attribute that describes how convenient and practicable the usage of a system is, that means if the system is able to support the user for a specific task in a best possible way. Only user-friendly and supportive systems will be accepted and can help the user to fulfil his or hers tasks in an easy and satisfying way. The subject-matter of this thesis addresses the difficulties and advantages that primary school children will have dealing with the XO respectively its user interface Sugar, thus the usability of Sugar in relation to its target group: children. In the following the meaning of the term *usability*, its history, usability engineering and usability testing in general and particularly with children will be explained.

3.1 From ergonomics to usability

In contrast to the assumption, that usability was coined during the *dot-com* boom in the late 1990s – it was the first time usability testing was used for commercial purpose –, its starting point is a little bit older [Spillers, 2007]. Already in 1857 the term *ergonomics* was suggested by Wojciech Jastrzębowski [Jastrzębowski, 1857] as an own scientific discipline in the domain of working environment, where the main focus should not be only on the technical system, but on the whole system, consisting of human, tools, assignment and environment. Ergonomics is defined as a:

> "Scientific discipline concerned with the understanding of interactions among human and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimise human well-being and overall system performance." [DIN EN ISO 6385, 2004]

Several decades later, during World War II, the first electronics and electrical systems were invented, that were controlled through user interfaces. The industrial psychologist John Flanagan discovered that the reduction of control elements (like buttons, knobs and switches) of these systems, could improve the users performance.

Until the early 1980s the term *user-friendly* was used when referring to usability, but gradually it "had acquired a host of undesirably vague and subjective connotations" [Bevan et al., 1991, p.1], and was later replaced by *usability*. Since then a multiplicity of usability definitions have been emerged, partially with different concepts. The first known definition is from 1984 and was established by Eason. He described usability as the deviation between the potential usefulness of a system – consisting of hardware, software and technical support (human) – and the level of which a user is able and motivated to use it [Eason, 1984].

According to the Swedish usability-expert Jakob Nielsen [Nielsen, 1993] and the computer scientist Ben Shneiderman [Shneiderman, 1992] usability is dependent on the attributes of a user interface that makes it easy to use. Nielsen is working in this field for more than 25 years and has elaborated five quality components to be able to investigate the usability of a system. These attributes are:

Learnability: How fast and easy can a user handle a system that he or she has never seen before, to accomplish basic tasks sufficiently?
→ If a user is able to use a system with all its features without reading manuals or being taught, just because it is as user-friendly and self-explanatory as possible, then the systems learnability is at its maximum.

Efficiency: How fast and efficient can a user perform tasks, after learning how to work with the system?
→ Only an efficient structured system allows the user to work fast and to accomplish the tasks in a pleasant way.

Memorability: How easily can a user reestablish the handling of a system?
→ The more the user remembers how to use the systems, the better he or she can resume the working process.

Errors: How often and how many errors do user make while using the system? How serious are these errors and how do users the recover from them?
→ A system should be as less error-prone as possible, therewith the user is able to work untroubled. If errors are in certain circumstances unavoidable, then the system should react as user-friendly as possible and point out an appropriate solution.

Satisfaction: How satisfied is the user with the system and its design?
→ Users should like using the system, only then they are willed to use the system again.

The International Organization for Standardization (ISO) narrows usability down as the

> "Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [DIN EN ISO 9241, 1998]

Summing up usability is an essential criteria for analysing a systems ability of being supportive for users to accomplish a specific task. It can be used during the development process of a system, and can also be the basis of an upgrading process. Only marginally should be mentioned that the next step after usability, may be *user experience*. It is the sum of usability, look and feel and poses the question if the user had a delightful experience as possible [Spool, 2007]. In the following standards and laws according usability evaluation will be explained.

3.2 Usability Heuristics

In this section the substantial standards of the ISO in relation to usability will be explained. These standards should help developers to respect users needs during the development process and create a convenient product.

Before 1990 several different usability guidelines and principles were deduced. But due to the fact that there were so many of them which had to be considered, a usability test could took very long to be concluded. For example the *Guidelines For Designing User Interface Software* from Smith and Mosier has 944 guidelines [Smith and Mosier, 1986]. Then in 1990 Rolf Molich and Jakob Nielsen developed a set of heuristics that was used by many usability evaluators thereon. The 10 heuristics from Molich and Nielsen are as follows [Nielsen and Molich, 1990]:

Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom: Users often choose system functions by mistake and will need a clearly marked "'emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention: Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition rather than recall: Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use: Accelerators - unseen by the novice user - may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognise, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

In contrast to the guidelines of Smith and Mosier, which were reviewed by many experts and are also based on results of research papers, Molich and Nielsen's heuristics were just deduced from a database of 249 usability problems that were collected from evaluations of 11 interactive systems. Bob Bailey warned in an *UI Design Newsletter* that:

> "These heuristics, which are widely used, have never been validated. There is no evidence that by applying these heuristics in the design and development of user interfaces that it will improve the interface." [Bailey, 1999]

The only research-based and international approved alternative for usability testing is provided by the ISO. The existence of these standards shows the importance of usability for the design of technical systems nowadays. The significant norms in relation to usability analysis are explained in the following subsections.

3.2.1 DIN EN ISO 9241

Part 110

In part 110 (part 10 until 2004) of the DIN EN ISO 9241 standard a set of usability heuristics in relation to the interaction - also referred to as *dialogue* - between users and information systems is presented [DIN EN ISO 9241, 1998, Part 110]. These seven "dialogue principles" are:

1. **Suitability for the task:** *Is the dialogue suitable for the user's task and skill level?*
   "A dialogue is suitable for a task when it supports the user in the effective and efficient completion of the task. In a dialogue which is suitable for the task, the user is enabled to focus on the task itself rather than the technology chosen to perform that task."
2. **Self-descriptiveness:** *Does the dialogue make it clear what the user should do next?*

"A dialogue is self-descriptive to the extent that at any time it is obvious to the users which dialogue they are in, where they are within the dialogue, which actions can be taken and how they can be performed."

3. **Conformity with user expectations:** *Is the dialogue consistent?*
   "A dialogue conforms with user expectations if it corresponds to predictable contextual needs of the user and to commonly accepted conventions."
4. **Suitability for learning:** *Does the dialogue support learning?*
   "A dialogue is suitable for learning when it supports and guides the user in learning to use the system."
5. **Controllability:** *Can the user control the pace and sequence of the interaction?*
   "A dialogue is controllable when the user is able to initiate and control the direction and pace of the interaction until the point at which the goal has been met."
6. **Error tolerance:** *Is the dialogue forgiving?*
   "A dialogue is error tolerant if, despite evident errors in input, the intended result may be achieved with either no or minimal corrective action by the user. Error tolerance is achieved by means of damage control, error correction, or error management to cope with errors that occur."
7. **Suitability for individualisation:** *Can the dialogue be customised to suit the user?*
   "A dialogue is capable of individualization when users can modify interaction and presentation of information to suit their individual capabilities and needs."

Part 11

Part 11 of DIN EN ISO 9241 describes the guidance on usability with no specific recommendations in terms of system attributes [DIN EN ISO 9241, 1998, Part 11]. Instead a definition of usability is given that was already presented in 3.1 (From ergonomics to usability). It is mentioned that the aim of the development and the validation of usable systems is that users can achieve a specific task in a specific context with effectiveness, efficiency and satisfaction. *Effectiveness* describes the accuracy and completeness of a specific goal that is archived. *Efficiency* is the relation between the effort and the effectiveness (thus accuracy and completeness) that is necessary to reach the goal. And *satisfaction* focuses on subjective criteria and refers to the users feeling of pleasure when the tasks is fulfilled.

3.2.2 DIN EN ISO 13407

The ISO standard DIN EN ISO 13407 pictures the user-oriented design of interactive systems. The main focus is on the holistic design-process, including technical, organisational, ergonomic as well as social and human factors. The main principles [DIN EN ISO 13407, 1999] of this norm are:

Involving the user in the development and design process.

Appropriate arrangement of functions between user and system, according to the users capabilities in relation to the system.

Iterative evaluation of design-layouts with users, to avoid non-fulfillment of requirements and for the optimisation of the system.

Multi disciplinary structure of a dynamical development team.

User-oriented or human-centred is here characterised by:

> "the active involvement of users and a clear understanding of user and task requirements; an appropriate allocation of function between users and technology; the iteration of design solutions; multi-disciplinary design." [DIN EN ISO 13407, 1999]

The appliance of these standards - which generally has to be adapted to the individual system (cf. 3.3.1. Usability Evaluation) - takes place during the so-called usability engineering process. This process will be explained in the next section. In section 5.1.1 (Criteria) the criteria that were used for analysis of SUGAR is illustrated.

3.3 Usability Engineering

Ideally, usability should be an issue right from the start during the development of a user-operated system. A methodical way to assure this is usability engineering, which is a sub-process of the development and design of a technical system. It complements the conventional engineering e.g. software-engineering with ergonomic prospects [Sarodnick and Brau, 2006, p.19]. These two processes (software-engineering and usability engineering) should be closely coupled during the production of a system, so that software engineers, designers and usability experts can cooperate, to achieve the best result. Furthermore the fields of cognitive- and work psychology should be considered and the systems target groups be involved in the process. According to Sarodnick and Brau this process can be divided into the following steps:

1. Analysis of the activity and the working environment
2. Analysis of the user group
3. Identification of requirements
4. Decision about functionality and derivation of an approach plan and an operational concept
5. Evaluation and improvement of the system during the development phase
6. Introduction and training
7. Further development

On the basis of these seven steps and two existent process models (the process-model of the DaimlerChrysler-Research and the usability engineering lifecycle from Mayhew [Sarodnick and Brau, 2006, p.82 ff]) Sarodnick and Brau deviated their own model of usability engineering that is presented in figure 3.1.

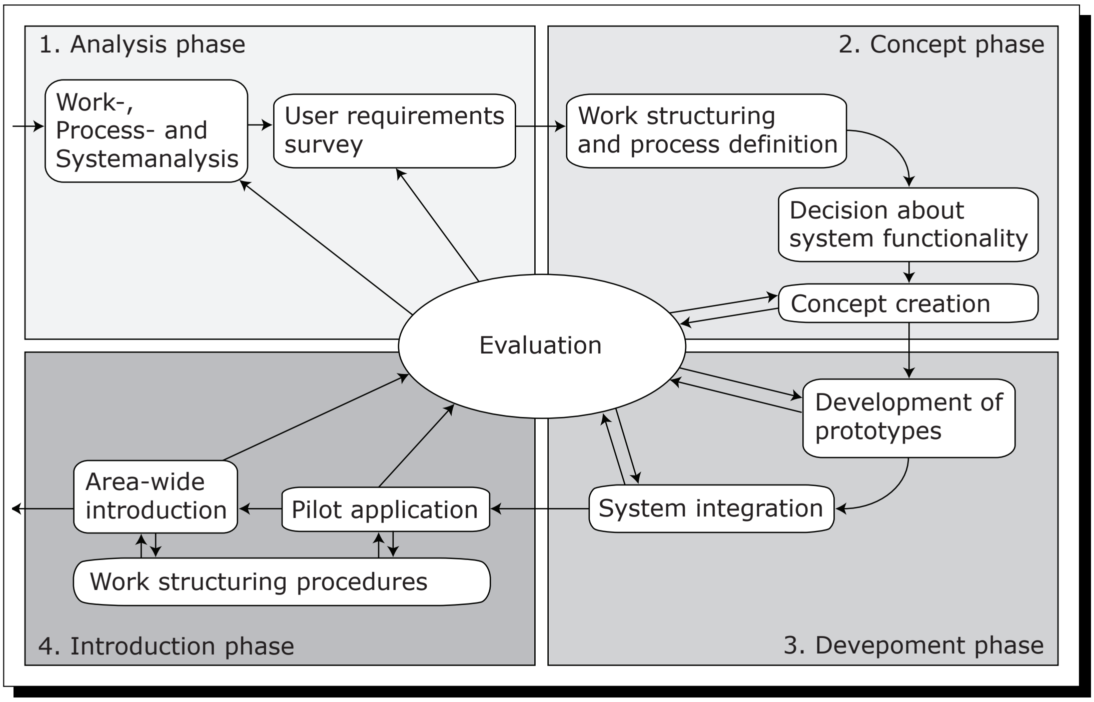

Figure 3.1: Usability engineering model from Sarodnick and Brau

During the first phase the operational procedures and user requirements are analysed to define a concept in the second phase. This concept is developed in the third phase (prototype) and introduced in the fourth and last phase. Throughout the entire process, regresses and transitions between these phases are possible, based on the centred usability activity. This usability activity or usability evaluation is a pivotal point in usability engineering and is explained in more detail in the following.

3.3.1 Usability Evaluation

Usability evaluation is the process of analysing a system, by applying usability methods, in relation to the systems fitness for purpose, that means if the system is able to support the user for a specific task in a best possible way. The methods of usability evaluation can be divided into two groups [Somervell et al., 2003]: *empirical* and *analytical* methods.

Empirical methods involve controlled tests accomplished by persons who are or even could be the users of the application. Specific aspects of the system are analysed to determine the efficiency of tasks that can be performed. The results must be interpreted, the occurred problems classified and solutions elaborated. Examples of empirical methods are usability tests, where users test a system while being monitored by usability experts and moreover questionnaires where users answer questions about the system.

Analytical methods on the other side involve the inspection of a system by usability experts, who identify the problems by means of a walk through process. This assessment is based on pre-defined criteria, but additionally the experience of the experts and their knowledge about the application area play also an important role. Analytical methods can help to interpret data from empirical methods. Examples are *heuristic evaluation*, where experts examine the system and rate its compliance with recognised principles rather heuristics and also *cognitive-walkthrough*, where a group of experts complete assignments on the basis of a task analysis and assess if all steps are feasible for a user.

Originally these usability methods were used at the end of or even after the development process, to measure the quality of the finished system and are part of the so-called *summative evaluation*. But nowadays the methods are more involved in the process of development to gather useful information and identify problems much earlier and pertain to *formative evaluation*. As already explained in 3.3 (Usability Engineering) the appliance of an usability analysis should be ideally performed right from the start during the development process.

For the evaluation of a system it is necessary to define suitable evaluation-objectives and -criteria respectively aims. On the basis of these aims the system can be analysed. A raw classification is provided by Gediga and Hamborg:

Which is better? Two systems are compared on the basis of defined criteria.

How good? This issue is related to specific features or attributes of a system and is mostly questioned at the end of a development process. To do this defined criteria are also necessary here.

Why bad? The classical variant of formative evaluation where the weakness of a system should be identified. [Gediga and Hamborg, 2002]

All three points of this classification can be applied almost unmodified to the examination of SUGAR. First, two systems (SUGAR and `Windows XP`) are being compared (*Which is better?*) – but heuristically, without making use of defined criteria – and secondly SUGAR is being tested to discover its advantages and disadvantages, when students in primary school interact with it (*How good?* & *Why bad?*).

However according to Sarodnick and Brau the illustrated ISO-standards (cf. 3.2 (Usability Heuristics)) are too general to use them directly as assessment criteria [Sarodnick and Brau, 2006, p.115]. Based on the aspects of the ISO-standards like suitability, conformity or controllability more precise project-specific criteria must be developed to analyse a systems usability. This criteria-finding-process is commonly done on the basis of experience of usability experts [Gediga and Hamborg, 2002]. The evaluation-objectives and -criteria regarding the examination of SUGAR in primary school will be presented in chapter 5 (Sugar in Action).

An important process during usability evaluation is the appropriate choice of an usability method, which is used to analyse a system based on defined criteria. Usability methods can help to improve a systems usability and usefulness. A sufficient package of the different important methods can be found in the *Usability Methods Toolbox* from James Hom [Hom, 2003] with additional further reading. The methods can be divided in three groups: *inquiry methods*, *inspection methods* and *testing methods* and are explained in the following.

Usability Inquiry methods

Usability experts talk to users to gather information about their likes, dislikes, needs and understanding of the system, observe the users while using the system in their working environment (not usability testing) or let them answer questions (written form or verbally). An overview about the common inquiry methods is shown in table 3.1.

Usability Inquiry methods	
Contextual Inquiry	Interview users in the field
Field Observation	Observe users in the field
Interviews and Focus Groups	Query experienced users with and provoke discussion about the system
Surveys	Ad hoc (interactive) interview with users - questions are asked and the answers recorded
Questionnaires	Written lists of questions are distributed to the users
Journaled Sessions	Users are observed and questioned while they experience the system
Self-Reporting Logs	Users log their actions and observations with the system in a journal
Screen Snapshots	The user takes screenshots while working with the system

Table 3.1: Usability Inquiry methods

Usability Inspection Methods

Usability experts (and sometimes users and developers) examine usability related aspects of a user interface. The most important inspection methods are described in table 3.2.

Usability Inspection Methods	
Heuristic Evaluation	Experts evaluate each element of the system against a list of accepted principles (heuristics)
Cognitive Walkthroughs	Experts *walk through* the interface as being a *real* user performing tasks
Formal Usability Inspections	Software resp. code inspection is adapted to encounter defects in the system
Pluralistic Walkthrough	Users, developers and usability experts step through a task scenario and discuss and evaluate the elements of interaction
Feature Inspection	The feature set of the system is analysed
Consistency Inspections	Consistence of common functions of multiple products from the same development effort are inspected
Standards Inspections	Product elements are inspected on the basis of industry standards
Guideline Checklists	Based on a guideline checklist the system is analysed to ensure that usability principles are considered (used during heuristic evaluation and consistency inspection)

Table 3.2: Usability Inspection methods

Usability Testing methods

Representative users accomplish typical tasks with the system. Experts observe and interpret the interaction between user and system to determine how well the user interface supports the user. Examples of testing methods are presented in table 3.3. Usability testing is described in 3.4 (Usability Testing) and the presented methods are explained in more detail in 3.4.3 (Method of collecting data).
Due to the fact that Sugar is being tested with primary school children, inquiry or inspection methods are not applicable, because inquiry methods are used when the users already worked with the system and inspection methods are typically performed by usability experts. This lead to the application of one or more usability testing methods. A validation of these testing methods is presented in 5.1.2 (Testing Method) to discover which method or methods are suitable for an usability test with primary school children.

At this point – respectively in the next subchapter – it is necessary to examine usability testing and what has to be considered while accomplishing usability testing, especially with children.

Usability Testing methods	
Coaching Method	The participant is allowed to ask questions and the tester serves as a coach
Thinking Aloud Protocol	During the interaction with the system the user is being asked to enunciate his/her thoughts
Co-discovery Learning	A couple of participants perform tasks together while being observed
Question-asking Protocol	The user is asked direct questions while interacting with the system
Performance Measurement	The performance is being measured, e.g. how long does it take the user to perform a certain task
Remote Testing	Tester and participant are separated: same-time but different-place or different-time and different-place
Retrospective Testing	The session is videotaped and can be reviewed with the participant after the test
Shadowing Method	An expert user explains the participant's behaviour to the tester - no think-aloud from the participant necessary
Teaching Method	Participants interact with the system on their own and should brief afterwards novice users how the system works

Table 3.3: Usability Testing methods

3.4 Usability Testing

With usability testing methods several features of a product can be measured, by asking representative users to perform predefined tasks. Measured are features like ease-of-use or the users perception about the product. According to Sarodnick and Brau usability testing is the most popular method to analyse a systems usability [Sarodnick and Brau, 2006, p.115] and Nielsen stated in 1993 that

> "Usability testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested." [Nielsen, 1993, 165]

Some usability engineers disagreed on the benefit of usability testing. Deborah J. Mayhew for example believes that usability tests raises the development costs [Mayhew, 1999]. But according to Nielsen these costs are reasonable and necessary, because users are not willed to learn difficult designs or want to work with slow systems [Nielsen, 1993].

Usability testing can be done *formally* in a usability lab (cf. 3.4.2. (Usability Lab)

– optionally be recorded with camera and microphone – or *informally* with paper mock-ups of the system. As already explained different methods can be applied (cf. table 3.3, but whatever method is used every usability test shares five characteristics [Dumas and Redish, 1999]:

1. Improvement of the products usability is the primary goal
2. The test persons represents the real users of the system
3. The test persons should accomplish real tasks
4. Every comment and reaction of the test persons should be observed and recorded
5. The collected data is analysed, the problems are inspected and suggestions to improvement the system are made

In the following the basic components of usability testing like target group, the usability lab and the most important methods according to Sarodnick and Brau [Sarodnick and Brau, 2006] are alluded.

3.4.1 Target group

A balanced assortment of test persons is important for usability tests. They should represent the end users as good as possible and not be familiar with the system. According to Nielsen and Virzi five to six persons are sufficient enough to discover up to 80% of the usability problems [Nielsen, 1993] [Virzi, 1992]. Sarodnick and Brau extend that the number of necessary test persons depends on the application area of the system [Sarodnick and Brau, 2006]. For example safety critical systems need more test persons, therefore all problems can be found. The target group for the Sugar-test is presented in 5.2 (Target Group and Setup).

3.4.2 Usability Lab

The equipment that is used to record the sessions plays also a decisive role during usability tests. Basically there are two kinds of usability labs: *static* and *mobile* labs. Static labs consists mostly of a soundproof control room with a one-way mirror and a test room. The proband processes the tasks in the test room while being observed from the control room. Portable equipment is part of the mobile lab. It is used when it is too tricky for the test user to reach the static lab or when it is necessary to test a system in a real environment.

A special form of usability testing is *remote usability testing*. No lab is needed, because the test persons perform tasks afar from the conductor. An example is a questionnaire located on a website. Remote testing provides no behavioural data of the test persons, only the activity can be recorded.

3.4.3 Method of collecting data

Several methods can be used during a usability test to gather information as already presented in table 3.3. All of them can additionally be video taped for *retrospective testing*. It is an auxiliary way to analyse the behaviour and the comments of the test person after they performed the tasks. Missed or rapid sequences can then be dissected unhurriedly. Nielsen thinks this method is to expensive and the cost should better be used for other tests [Nielsen, 1993], but it is certainly a useful method when the observer is not able to write down everything during the test. The test person can also be asked to attend afterwards for a *video feedback*, where the proband can explain her or his activity while performing the tasks. An alternative to video recording are *log-files*, which are compiled during the test and record every activity-sequence that is performed. But in this case no behavioural data of the test person are collected.

The *thinking aloud method* is a process of collecting conscious cognitions and emotions that accompany the behaviour of the test person [Gediga and Hamborg, 2002]. During the test the proband is asked to enunciate his or her thoughts and performance. A disadvantage of this method is its slowness, because the user needs more time to perform a certain task, while articulating his or her cognition. *Co-discovery learning* can be applied where two test persons perform the tasks together. A natural discussion can emerge about what to do next and usually more comments are spoken out compared to the thinking aloud method [Sarodnick and Brau, 2006].

Commonly the test user is doing the tasks on his or her own, without additional help. The method where the interaction between proband and controller is explicitly requested is called *coaching method*. The controller gives any needed assistance if the test person needs help.

Performance measurement is used if time- and error-data needs to be measured to detect strengths and weaknesses of a system [Sarodnick and Brau, 2006]. It is particularly used when two or more systems are compared. Sardonick and Brau provide an overview what can be measured with this method [Sarodnick and Brau, 2006, 165]. For example the time that is needed to accomplish a task, the number of tasks that can be done in a specific time, the number of errors or the relation between positive and negative feedback.

Whatever method or methods are used during usability tests, it is important that the test users are not deviated by the environment or the observer. Moreover the observer should not direct the testers attention to any part of the screen. Both could lead to falsification of the results.

The method or methods will be used during the Sugar-test in primary school is explained in section 5.1.2 (Testing Method). The next section is addressed to characteristics in relation to usability testing with children and explains what needs to be considered during the accomplishment.

3.4.4 Usability Testing with Children

Performing usability tests with children is a special case. Several aspects have to be considered and the tests require diverse and additional preparation - in contrast to usability testing with adults. In the following the particularities of usability testing with children will be described.

Hanna et al., usability engineers at Microsoft and often cited, depict a general overview about the field of usability testing with children. They describe what is important and has to be considered when perform these tests in their report *Guideline for Usability Testing with Children* [Hanna et al., 1997]. This guideline - evolved during their research, mostly usability lab tests - allows children to participate in traditional software usability tests and will also be considered during the Sugar-testing phase with primary school children. Before this guideline is presented, the age groups with their divers behaviour are explained. Afterwards only the rules correspondent to the target group of the Sugar-test will be presented.

The age of the children that should take part in an usability test is an important point. Children with the age of two react and perform different compared to children who are already 14 years old. Hanna et al. describe three different target age ranges: preschool-aged children (2 to 5 years), primary-school-aged children (6 to 10 years), and middle-school-aged children (11 to 14 years). The also indicate that this division is erratic and the children's behaviour can overlap. Preschool children between 2 and 5 years "should be allowed to explore the computer according to their own interests and pacing instead of performing a series of directed tasks" [Hanna et al., 1997, p.10] and they have often difficulties to express their likes and dislikes. Children in primary school (6 to 10 years) "are relatively easy to include in software usability tests" [Hanna et al., 1997, p.10]. Because of their school-experience they are able to concentrate on a task and follow instructions of an adult. While sitting in front of a computer they are generally not aware of being watched. They will answer questions willingly and are motivated to try new things out. Furthermore children in this are range "will develop more sophistication about how they can describe things they see and do" [Hanna et al., 1997, p.10]. Hanna et al. made also a differentiation for primary school children: six- and seven year old children may be a little shy when answering questions and won't have much experience with computers. Ten-year-old children have probably more knowledge about computers and are able to criticise the system. Beyond that they point out that middle school children aged from 11 to 14 years may "bring a very high level of computer expertise, or distinct expectations for what they will be doing, to a usability session" [Hanna et al., 1997, p.11].

The SUGAR tests were performed with children from primary school and in subchapter 5.2 (Target Group and Setup) this decision is described in more detail. The guidelines from Hanna et al. can be divided into four parts: planning, introduction, testing and finishing up and will described in the following.

1. Planning

- The testing environment should be as child-friendly as possible (e.g. colored walls, posters, etc.), so that the child feels as comfortable as possible. On the other side there should not be anything that distract the child, like toys or paint utensils, so he or she can concentrate completely on the task. Best the testing is done in a place that the child already knows, like his or her classroom or home.
- If the child has already used a computer, the input device should be selected according to the child's experience.
- Microphone(s) should be placed so that even their low voice can be recorded and camera(s) should not face them directly, because that could irritate them.
- Testing time should be between 30 and 60 minutes, otherwise the children can get too tired and inattentive. According to Hanna et al. children's concentration starts to decrease after 30 minutes [Hanna et al., 1998]. The tester should also rest enough time in between, because testing with children can be very exhausting, especially when observing, motivating and providing feedback at the same time.
- Children with too much computer experience should not be included, if they are not especially the target group. They would not provide the necessary data. On the other side, children with no computer experience are also not advisable, because it can take long until they are familiar with the computer.
- A letter of agreement is necessary to inform the parents of the children and get their approval, that the child is allowed to participate in the test.

2. Introducing

- Before the testing phase it is advisable to establish a relationship with the children by initiating a small talk about their interests, hobbies etc.
- The children should understand that it is not about testing them, but rather their help is needed to improve the software.
- The child's expectation should be clarified before the test. If they expect to play a finished game, but the prototype consists only of static pictures, it could happen that they will be disappointed during the test.

- For some children in the age of 7 or 8 it may be disturbing when leaving them alone in a test situation and giving them instructions via a speaker. So it is better to be present during the test.

3. Testing

- A warm-up phase at the beginning of the test phase with every child is helpful to get comfortable with the product.
- Tasks should be separated into smaller pieces, that they are more comprehensible and understandable for the children. It is also helpful that they express the task in their own words to check if they understood the assignment.
- Questions of the children should not be answered directly. Instead they should be motivated to find out on their own. An example could be: "Should I click on this icon?" – "What do you think?". A prepared script with hints to support the children can serve as a tracking-tool to detect what kind of help was needed during the test.
- Children should get direct requests, instead of asking them if they want to do something. Otherwise they have the possibility to decline the task.
- If the childrens' motivation decreases, they should be carefully reminded to pay attention. The tester can pretend to need help from the child to get their focus back on the computer.
- After a testing period of 45 minutes the children should get the opportunity to take a short break.
- Children should get sufficient positive feedback, so they do not get the feeling of failing to solve the tasks.

4. Finishing up

- During the tests it is helpful to keep an eye on any sign of engagement from the children, like laughing or frowning. These sign can be more meaningful as the spoken responses from the child.
- If a questionnaire with ratings after the test is performed it is necessary to clearly mark the different values (e.g. a thumb-up symbol instead of "I liked it.")
- After the whole test is finished it would be satisfying for the child to explain them how important and helpful their help was.

These aspects were be considered during the SUGAR-test in primary school and its realisation is explained in chapter 5 (Sugar in Action). The next chapter of the thesis is addressed to the comparison of SUGAR and `Windows XP`.

4 Sugar vs. Windows

"A picture is worth a thousand words. An interface is worth a thousand pictures."

Ben Shneiderman, 2003 [LukeW Interface Designs, 2006]

Before testing SUGAR in primary school to gather its assets and drawbacks in relation to the usage in children's hands, it will be compared to another already established system. The decision with which other system SUGAR should be compared is simply based on the operating system statistics from W3Schools. In their statistics from July 2009 the Windows family counts for over 90% and `Windows XP` is since September 2003 the most popular operating system with currently 67.1% [Network Solutions, 2009]. Accordingly, `Windows XP` is best suited for a comparison because it is the most widespread operating system respectively GUI. This comparison will highlight the differences between an *old* (`Windows XP`) and a *new* (SUGAR) system.

In the following only the interfaces of both systems are compared and all operating system relevant functions are ignored. To be more precise, the fields that will be examined in this chapter are the structure of both GUIs and their interaction and communication possibilities. The status of both systems is the following:

Windows XP: Windows XP Professional, Version 2002 with Service Pack 3
Sugar: Stable build 767 (8.2.0)

No extra software or drivers were installed, the systems are only updated regularly to guarantee best possible performance and features.

This comparison is made heuristically, more precisely, without the useage of defined criteria, only rated respectively benchmarked by myself based on more than 20 years of experience working with user interfaces.

4.1 GUI

In this section the major differences between SUGAR and `Windows XP` concerning their interface structure will be described. Each of the following subsections is addressed to one specific distinction.

4.1.1 Desktop Metaphor vs. Zoom Metaphor

The GUI of `Windows XP` (cf. figure 4.1) consists of two major elements. The big main area is called the *desktop* where files, folders and shortcuts can be stored and the applications are presented in their own window (cf. 4.1.4 (Multiple vs. one Window)). And the second element is the taskbar, which is placed at the bottom and described in more detail in the next subsection. The user can create, delete, copy, paste, rename and arrange files and folders (and shortcuts to files and folders) on the desktop (and anywhere else on the hard-drive – cf. 4.1.5 (Explorer vs. Journal)). In figure 4.1 shortcuts to e.g. *My Documents*, *My Computer*, *My Network Places* and the *Recycle Bin* are shown. The window of a program can be arranged, minimised and maximised on the desktop (cf. 4.1.4 (Multiple vs. one Window)). The desktop is always present and only not in sight, thus in the background, if a program window is maximised.

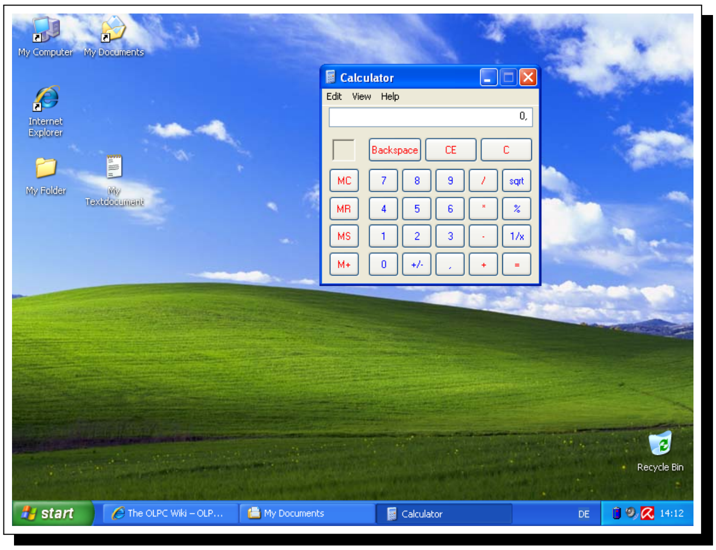

Figure 4.1: `Windows XP` desktop

SUGAR has a completely different approach. It is not an image of a real desk and has no desktop like `Windows XP`. As explained before (cf. 2.3.3 (Zoom Levels)), the structure of SUGAR is based on the so-called *zoom metaphor* and is related to its networking possibilities. The different zoom levels describe a kind of *ride* from the network level to the actual activity that is in use. The level without a zoom shows the *neighborhood* with all other computers and access points in range. The first zoom level shows the *groups* the child respectively the XO takes part in. The next level shows the *home screen* as displayed in figure 2.8. This is the only view that looks as similar as possible to a *common* desktop of an user interface. Here activities can be launched, system settings accessed and the computer restarted or shut down by using the appropriate icon in the SUGAR-ring. This ring is compared with the `Windows XP` Start Menu in 4.1.3 (Start Menu vs. Sugar-Ring). The maximum zoom level shows only the activity that was launched last. Thus it is not possible to arrange different application windows on the screen like in `Windows XP` (cf. 4.1.4 (Multiple vs. one Window)).

SUGAR has also a kind of menu bar, the so-called *Frame*. Within the Frame it is for example possible to switch between the different zoom levels, buffer any kind of object for later usage or close the actual activity. The Frame will be compared to the `Windows XP` menu bar in the next subsection.

Additionally, SUGAR focuses more on what the user – primary children – can do (write, draw, create) instead of what they have (documents, folder). That means `Windows XP` focuses more on objects, like on a real desk and supports the user to create different kind of documents (text, spread sheets, etc.). SUGAR on the other side is more activity oriented. In the centre of SUGAR is the process of creation, represented by several different activities for writing a story, drawing a picture or letting a turtle walk around by *writing* an appropriate program code. This can be done alone or with other user that are using other XOs. And so the creation-process can be shared and the users have the chance to learn from and support each other. More about *verbs* and *nouns* is explained in 4.1.7 (Verbs vs. Nouns).

4.1.2 Taskbar vs. Frame

By default the *taskbar* in `Windows XP` is at the bottom of the screen (cf. figure 4.1), but can be docked to the top, left or right side. Even its size can be altered. The taskbar consists of several elements. Leftmost is the green start button, that activates the Start Menu. Through the Start Menu applications can be launched, documents found and commands executed (cf. 4.1.3 (Start Menu vs. Sugar-Ring)). The largest area of the taskbar shows the applications that are opened. Each application window has its own entry (cf. 4.1.4 (Multiple vs. one Window)). In figure 4.1 three programs are opened: the Internet Explorer, the Windows Explorer and the calculator. If a program window is minimised it remains in the taskbar and can be maximised by clicking on it. Rightmost is the system tray with a clock and launched services like an activated virus scanner, firewall or available system updates. It is also possible to add several *toolbars* to the taskbar. In figure 4.1 the *language bar* is activated to change language setting. Other toolbars that can be added are for example a *Quick Launch* bar with shortcuts to favourite programs or to favourite internet links.

The pendant in SUGAR is called the *Frame* (cf. figure 2.13) as already explained in subsection 2.3.4 (The Frame). The Frame is divided into four areas: the *objects*-area on the left for temporary storage of objects (comparable to the clipboard in `Windows XP`), the *places*-area at the top with the search bar, the zoom buttons to switch between the different views and icons of all opened activities, the *people*-area on the right where all other users are shown that are collaborating in the current activity and the *actions*-area at the bottom where invitations and system notifications are being displayed.

4.1.3 Start Menu vs. Sugar-Ring

By clicking on the green start button in `Windows XP` – by default in the lower left corner – the *Start Menu* is activated, as shown in figure 4.2. The *Start Menu* contains a list of all installed applications called *All Programs*, so that these can be found and launched easily. Shortcuts to often used programs can be placed directly in the first level of the menu, for example the Windows Media Player as displayed in figure 4.2. Additional to program shortcuts, links to the users documents, like *My Documents*, *My Pictures*, *My Music*, are embedded in the Start Menu, equal to the shortcuts on the desktop (cf. 4.1.1 (Desktop Metaphor vs. Zoom Metaphor)).

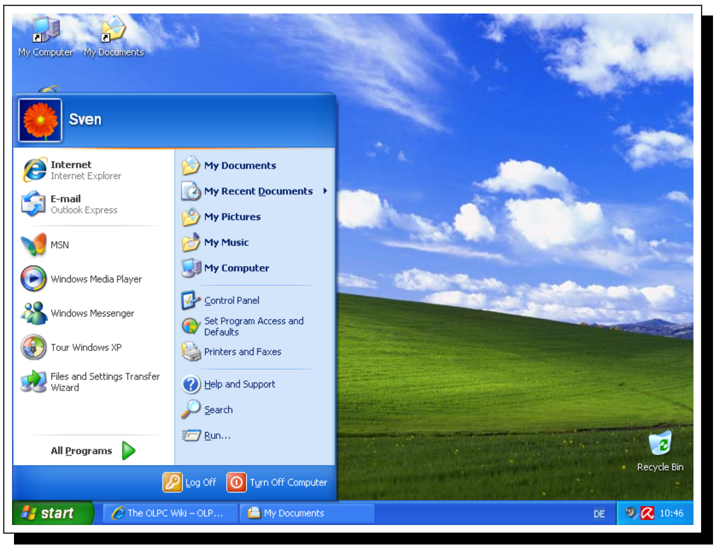

Figure 4.2: The `Windows XP` Start Menu

The Start Menu also contains some other functions. System settings, like user accounts, network connections or printer setup can be changed within the *Control Panel*. *Program Access* and settings for printers and faxes can also be launched from the Start Menu. All files on the computer can be searched via the *Search*-function. A *Help and Support* area can give useful advices to users that are stuck or need extra information about the system. Programs, documents or web sites can be launched without searching the appropriate shortcut in the Start Menu, by entering the exact command after clicking on the *Run* field. It is noticeable that several functions within the Start Menu can be reached via different ways. For example the entry for printer settings is directly in the first level of the menu, but also within the *Control Panel*. Moreover shutdown, restart, hibernation, standby and changing user accounts is possible via the menu.

The SUGAR-ring, displayed in figure 2.8 and already explained in 2.3.3 (The Home Screen), can be compared with the *Start Menu* of `Windows XP` because it contains icon-shortcuts to the installed activities, system settings and the possibility to shut down or restart the system. By default, all installed activities are arranged around the XO user icon, but can also be adapted to the user's needs, so that only icon-shortcuts to favourite activities are displayed. The current activity appears beneath the XO user icon – in figure 2.8 it is the Record activity. In an alternative view-mode the icons can also be arranged individually on the screen. A list view of all installed activities is the third possibility to browse and launch any of them.

Here it has to be mentioned that icons of all opened programs appear only in the upper part of the Frame, in order that switching between activities is possible (cf. figure 2.13 and 2.3.4 (The Frame)).

4.1.4 Multiple vs. one Window

As already explained before, each application has its own window in `Windows XP`. The size of these windows can be altered, they can be maximised or minimised to the taskbar (cf. 4.1.2 (Taskbar vs. Frame)) and they can be arranged so that more than one window can fit on the screen. This feature can be helpful while working with different applications at the same time, for example programming a webssite in one window and previewing the page in a browser window.

In contrast to this multitasking approach SUGAR uses a fullscreen activity view "that focuses the children's energies on one specific task at a time" [OLPC, 2009]. Although it is possible to have multiple activities – or instances of activities – opened, only one activity can be the active one.

4.1.5 Explorer vs. Journal

Every installed operating system is able to store and open files, so it is possible to stop working on a document anytime and resume working on it later. Within `Windows XP` the user usually has to take care about the filename of the document and the place where to save it on his or her own. Files can be stored anywhere on the hard drive, which can get very confusing when handling a huge amount of files. Some applications create even an own folder by default. Appropriate naming conventions and an efficient file structure is needed when dealing with many files to ensure that these files can be found later.

By default it is not necessary to store any kind of file respectively object within SUGAR. It contains a file manager, called *the Journal*, that stores automatically every object that is created with any activity. The user does not has to take care of a file or folder structure on his or her own. The Journal is the only activity that is launched automatically when the systems starts. By means of a search field it is possible to find existing objects in the Journal. The Journal is described in more detail in 2.3.5 (The Journal) and is illustrated in figure 2.14.

4.1.6 Applications vs. Activities

In relation to the installed programs and the documents that can be created, two things are significantly different between both systems: SUGAR is Open Source in contrast to `Windows XP` and their basic elements have a different meaning and function.

The SUGAR-interface is, like its foundation, the Linux distribution *Fedora*, open source. This means that all components and their source-code are available for free and can be used and altered by everyone. This has the advantage that many people can work on and improve the system and its programs and publish new features and activities. Furthermore the system can be adapted to the needs of the user, for example a school could include their own logo in the interface. New activities can be downloaded and installed easily with the help of the Browser activity. `Windows XP` on the other hand must be purchased, as well as some of the available software. But there are also many applications free of charge, partially even open source. Nearly every priced program has a free counterpart with matchable features. For example the English version of the graphical software *Photoshop* costs in the up-to-date version around $699 and the open source alternative called *GIMP* is available for free.

In `Windows XP` the different tools that can be launched, like a notepad or calculator are called programs or applications. Within Sugar the tools that can be launched are called *activities*. The items that can be created within `Windows XP` are called files or documents, like elements on a real desktop. The complement elements in Sugar are called *objects* and the process of creating and saving these objects is done by the system automatically (cf. 4.1.5 (Explorer vs. Journal)). "OLPC aims to provide a platform which encourages expression through creation. In support of this idea, activities – not applications – provide the main tools through which objects are created" [OLPC, 2009]. According to Martinazzo et al. "this is more than a naming convention; it is an educational paradigm" [Martinazzo et al., 2008, p.845] and they quoted OLPC "activities are distinct from applications in their foci – collaboration and expression – and their implementation – journaling and iteration" [Martinazzo et al., 2008, p.845].

4.1.7 Verbs vs. Nouns

`Windows XP` and all other traditional desktop interfaces focus on nouns because the dominant area of the screen shows documents, folders and applications. Only the small menu bar each application has embedded at the top is represented by verbs: for example open, close, insert and edit. This is due to the fact that all common GUIs represent a real desktop with papers and folders, as already mentioned and this makes total sense to an office worker or executive, but it may not appropriate for children. According to Thornburg children are more interested in what they can do (verbs) instead of what they have (nouns) [Thornburg, 2008, p.5]. OLPC considered this during the development of Sugar and substantiated that

> "Activities are verbs. As such, the phrase '⟨activity⟩ with my friends' should make sense. For instance 'draw with my friends,' 'browse with my friends', 'chat with my friends' and 'edit text with my friends' all make much more sense than 'text editor with my friends.' [...] Treating the activity as an action (verb) and not as a thing (noun) maintains the interaction model that the laptop tries to embody." [OLPC, 2009]

4.1.8 Icons and Naming

Both systems use unique icons for their applications. They appear on the desktop or home screen, in the *Start Menu* or the *Frame* and in the *Windows Explorer* or the *Journal*. The main difference between the icons of both systems is the number of colours. Whereas icons in `Windows XP` can be multicoloured, the Sugar icons are only two-coloured: white and grey when the activity is inactive or according to the user colours (cf. 2.3.3 (Zoom Levels)) when the activity is loaded. This limitation within SUGAR is related to the display of the XO. As already mentioned in 2.2.1 (Display), the display is also readable in bright sunlight, if switched to monochrome mode. The combination of icon colours that are selectable have enough contrast to be clearly legible even in this monochrome mode. Additionally in relation to the colour limitation of the XO's monochrome mode, the icons require a minimal design, therefore they are read- and understandable.

No naming or icon conventions exist for both systems and each developer or company can name and design an icon individually. But OLPC asks "whenever possible, activities should be named with descriptive verbs, or suitable pseudo-verbs, in order to emphasise their function as things you do" [OLPC, 2009].

4.2 Communication

Communication plays a vital role in our society these days and in the last decades several new communication possibilities were invented and changed the way of interaction, for example the telephone, the mobile phone, the computer and the internet. The need of online communication and activities rises steadily, especially among children. In this context Lenhart et al. investigated the communication behavior of U.S. teens:

> "Some 93% of teens use the internet, and more of them than ever are treating it as a venue for social interaction – a place where they can share creations, tell stories, and interact with others." [Lenhart et al., 2007, p.2]

Additionally, they found out that

> "[...] 64% of online teens ages 12-17 have participated in one or more among a wide range of content-creating activities on the internet, up from 57% of online teens in a similar survey at the end of 2004." [Lenhart et al., 2007, p.2]

In the following the communication possibilities of both SUGAR and `Windows XP` will be examined.

In order to communicate with others when working with `Windows XP` or SUGAR a functional network connection is required. The technical requirements should be neglected as far as possible, due to the fact that the focus is on the systems GUIs. Both are able to connect to an insecure network automatically. Is the meshwork otherwise protected and a user identification necessary the user has to intervene. However, SUGAR has the edge over `Windows XP`, because it offers the advantage of being able to connect to other computers with the same system.

Both systems provide possibilities to communicate with others and have pre-installed chat programs that allows the user to communicate with others in realtime. Video chat software for free is available for both systems. In terms of communication via e-mail, SUGAR has no mail application installed. But since Microsoft stared the first web-based e-mail service called *Hotmail* in 1997 no extra program needs to be installed on any system. The *Gmail* activity that can additionally be installed on SUGAR is only a modified version of the Browser activity that opens `gmail.com` as the homepage.

SUGAR provides up to now a unique opportunity of inviting other XO users in network range to collaborate in any activity that is installed on all participating systems. One or more users can be invited to be creative together. The invitations appear in the bottom part of the Frame (cf. 2.3.4 (The Frame)). The resultant interaction possibilities are described in the next section. As stated before OLPC plans to develop *bulletin boards* for each zoom level of SUGAR. Such a bulletin board system would supply an additional communication possibility because the users would also be able to discuss their collaboration.

4.3 Interaction

Interaction possibilities are present in both systems, but SUGAR provides more alternatives. Games like *solitaire* (card game) and *minesweeper* (logic game) are part of `Windows XP`, but are not as educational as the games in SUGAR. *Memorize* for example is a mathematical version of the game *memory*, in which the user has to find the correct result-pairs of arithmetic problems.

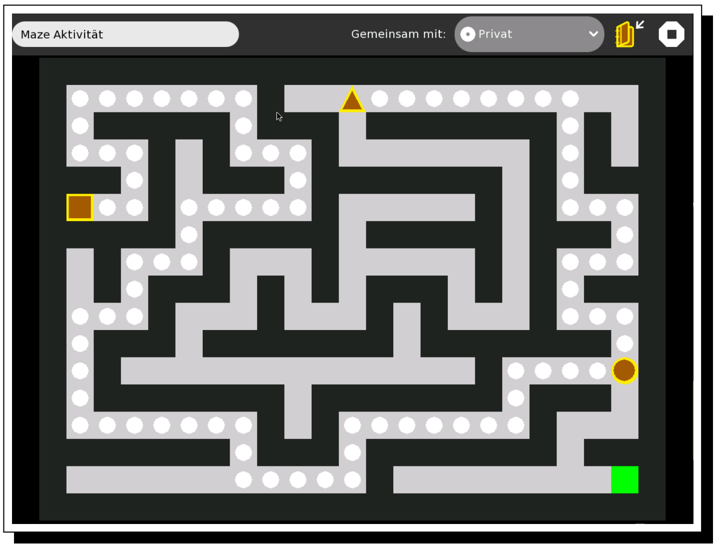

Figure 4.3: SUGAR Maze activity

Learning software – nonexistent in the standard installation of `Windows XP` – is also part of the SUGAR GUI. In the basic version of the GUI four different activities are available to learn the principles of programming. *Scratch*, for instance, is an easy-to-learn, multimedia programming language, where colourful command-blocks can be snapped together to create interactive games and animations.

Sugar also provides some activities to produce music. Four music activities are available for jamming, composing, and synthesising. Another audio example is *Speak*, which is an animated face that speaks whatever the user types in. An example of a multiplayer game is *Maze*, a labyrinth game, where up to three users can play against each other – at one laptop or via network. A screenshot is presented in figure 4.3.

A full list of all available activities can be found on `http://wiki.laptop.org/go/Activities/All`.

Besides this – as implied in the 4.2 (Communication) – all Sugar activities can be used with other XOs, to create objects together, regardless if the XO is connected to other XOs directly or via a network. The Sugar users can invite each other and create objects together. For example, several users can cooperate to draw a picture or write a text. This opportunity allows users to share their creations and ideas, teach each other and get connected. `Windows XP` provides no similar opportunity.

4.4 Summary

To sum up, `Windows XP` and Sugar are two very different user interfaces. `Windows XP` is based on the *desktop metaphor* (representing a desk) and Sugar on the *zoom metaphor* (different zoom levels show the different levels of the GUI). The aim of `Windows XP` is to support the user doing different tasks originally from office workers. Sugar supports the user – primarily children – to create objects and share the results with others. Communication is as fundamental as the process of creation, because with Sugar the user respectively the users are able to create objects together, while sitting in front of different laptops. Due to the fact that Sugar has more programs or activities pre-installed (also many other can be installed later), it offers more interaction possibilities as `Windows XP`. Furthermore Sugar and all its activities are free and open source. Because of the simplification of icons, but nevertheless colourful environment Sugar looks more playful and child-friendly than `Windows XP`.

Besides the fact that Sugar was developed especially for children, in contrast to `Windows XP`, it seems to be – from a theoretical and subjective point of view – better suited to support children. Sugar can even motivate children to work more (cf. 2.4.1 (Field reports)). But there are also some critical points that have to be considered. As mentioned in 2.4.3 (Interim balance), is Sugar with its playful design able to prepare children for their later working environment? Another point is the Journal. Is the Journal a lasting file management solution? What if a child does not remember when she or he has drawn a picture, but wants to find the picture again at a later time? Without knowing the date, the whole Journal entries containing drawings have to be searched. Maybe with another system, the child would have created an own folder for that picture. On top of that the XO is not the fastest computer, but according to OLPC its power is sufficient enough for the activities that are installed. But the demands of a child may change the older he or she will get and the studies and reports have shown that the children already were unsatisfied with the processing power of the XO. The test with primary school children partially considers the mentioned aspects and the results of the test are presented in 5.4 (Observations and Analysis).

The realisation and the results of testing Sugar with primary school children will be described in the next chapter.

5 Sugar in Action

"As the children grow and pursue new ideas, the software and the tools need to be able to grow with them and provide a gateway to other technology."

OLPC [OLPC, 2009]

In the previous chapters the OLPC project and their educational tool, the XO-1 with the new developed SUGAR-interface was introduced. After touching the subject of usability and usability testing and what has to be considered especially when working with children, the GUIs of `Windows XP` and SUGAR were compared. Based on defined criteria and the appliance of adequate usability testing methods, SUGAR was tested in primary school to identify the vantages and difficulties when children handle the XO respectively SUGAR. The content of this test were several task that the children had to perform.

In this chapter the methodology respectively criteria and the usability methods that were applied during the test will be explained. Furthermore the structure of the school-test is described including task, equipment and a test group description and finally the collected data will be analysed.

At first the criteria that were the basis of the SUGAR test in primary school, the tasks that were performed and what difficulties were expected are explained.

5.1 Methodology

In order to test any kind of system that is equipped with an user interface, to analyse its level of usability – in general or in a specific context – appropriate criteria need to be defined. In chapter 3 (Being usable) particularly in section 3.2 (Usability Heuristics) the usability heuristics from Molich and Nielsen and from the International Organization for Standardization (Part 110 and Part 11 from DIN EN ISO 9241) concerning usability were presented. Examining all of the presented heuristics would be to complex and go beyond the scope of this thesis. And according to Sarodnick and Brau these standards are too general to use them directly as assessment criteria (as already described in 3.3.1 (Usability Evaluation)), that means the presented aspects serve as a basis for the development of more precise and specific criteria. They where applied during the SUGAR usability test with primary school children – the target group is described in more detail in section 5.2 (Target Group and Setup) – and the analysis phase of the collected data.

5.1.1 Criteria

In the following the defined criteria are explained.

Consistency

Is the user interface in all its particulars consistent? Have all SUGAR-activities the same layout, design and elements. For example are all menu bars – if existing – at the same position. Is the German-translation complete?

→ Especially for children it is important, that such a tool is consistent, so that working and learning are possible without distraction, misunderstanding or uncertainty. Due to the fact that SUGAR and all its components (also the programs respectively activities) are open source, a consistent structure of all elements may be nonexistent, because so many different people can work on the system and its activities,

Self-Descriptiveness

Is it at any time obvious and clear to the user which dialogue they are in, what they can do next and how to perform the next step?
→ The system should be able to describe itself. At any time it has to be clear what possibilities the user has to go and how to proceed. Otherwise the user can get impatient or bored. SUGAR has many different elements, like symbols, icons, partially menu bars, the different zoom levels and the Frame (cf. 2.3 (Sugar)), all these components should be designed in a way so that the users is able to understand what options she or he can take.

Conformity with user expectations

Does really that happen what the user estimates, when using the system? Does e.g. clicking a button, really provoke what the user predicts?
→ For example, by reason that within SUGAR as many symbols and icons as possible were use, to avoid written language whenever possible, these symbols need to be understandable and not ambiguous. Because SUGAR was developed for children all over the world, in countries with different cultures, the used symbols could be misinterpreted by some users. But it is significant for an usable system, that the expectations of the user what happens next are met by the system and it will be examined if the symbols and icons are understandable for primary school children at a German school.

Ways of operating

Are there different ways to reach a goal or leads only one way to the result?
→ As aforementioned SUGAR has many different symbols and elements. They could provide different ways to reach a goal or perform a specific task. Too many ways could confuse the user instead of providing support.

Satisfaction

Is the system able to support the user in a satisfying way, or are there situations where the user can get frustrated, because of annoying events?
→ In both studies that were presented in section 2.4 (XO-Experience) an inaccurate and jumpy cursor was mentioned by the children, while working with SUGAR. Sure enough this problem could also be attributed to hardware problems, but has to be

considered as well, because this could lead to frustration and slipping motivation. Not only this unpredictable event could lead to dissatisfaction, also unclear structure of the system could be a trigger as well.

Curiosity

To what extent is Sugar able to arouse the users curiosity and spark their initiative to get creative?
→ This criterion is relatively hard to measure, because it is not foreseeable which activity or task could stimulate the user to get creative and explore the interface on his or her own. Nevertheless it is at least advantageous, if the system is able to awake the fantasy or creativity of the user to try new things out, and so the collected data from the tests was also analysed according to this criterion.

5.1.2 Testing Method

In subsection 3.3.1 (Usability Evaluation) the need of applying one or more adequate usability methods was already explained, if it is a matter of measuring the usability of a system. The available methods can be divided into three groups: *usability inquiry methods*, *usability inspections methods* and *usability testing methods*. Due to the fact that the examination of Sugar is a summative evaluation, because it is a question of analysing an already completed system, and furthermore inquiry methods require users that are already familiar with the system and during the application of inspection methods the system is normally tested by usability experts, consequently one or more usability testing methods need to be used. To discover which method or methods are appropriate to analyse the results of the Sugar test in primary school they have to be validated.

Due to the fact that tester and participant are in the same room at the same time *remote testing* is not applicable. *Teaching method* and *co-discovery learning* are also not appropriate, because only one student is tested at a time – otherwise they may distract each other and are not able to focus on the tasks. By reason that the children should not be overstrained during the tests, the *thinking aloud*- and the *question-asking protocol* are used neither. Both methods could also confuse the children and interfere the procedure. Although the session will be videotaped, the *retrospective testing* is also not an appropriate method, because children's concentration starts to decrease after 30 minutes and if the whole session is being analysed with the children afterwards – and it should be done directly afterwards so they may remember what they did and why they did it – a session could only lasts 15 minutes, which would be too short (cf. 5.1.3 (Tasks)). By reason that no expert Sugar-user takes part in the sessions, the *shadowing method* is also not useful for the analysation. The remaining two methods are the *coaching method* and *performance measurement* and are explained in more detail in the following.

Performance measurement

The goal is to examine SUGAR according to its difficulties and vantages when primary school children handle the XO respectively SUGAR in respect to usability. A usability testing method need to be applied that allows to analyse the systems ability of supporting the user while working with it and correspondingly a tasks list was elaborated that should be performed by the children (cf. 5.1.3 (Tasks)). The *performance measurement* method is a suitable method that allows to analyse the accomplishment of predefined tasks by the user. This method can not only be applied to analyse the length of time a user needs to perform a task, but also if she or he is even able to perform a task and also how the task was performed. If the user had problems and needed help or which tasks were understood and in which way the system is able to support the user.

Coaching method

Given that is was not foreseeable how much experience the users respectively the children already have dealing with a computer system – the target group is explained in 5.2 (Target Group and Setup) – and that the GUI was not be introduced to the children before the meetings, it is necessary to be able to support the children during the test if they need help. These problems that could occur during the test may as well be helpful to examine how well the system can be handled by the children. When the children are not able to perform a certain task on their own, it is beneficial to help them, so that they can finish the tasks and keep their motivation. But instead of giving direct support to the children how to proceed or what to do next, they should be motivated to find out themselves, as proposed by Hanna et al. (cf. 3.4.4 (Usability Testing with Children)). For example, if they are not sure which icon they should click to open a specified activity and they ask if some icon is the correct one, it is better to let them find out themselves. According to this a usability testing method where the participant is allowed to ask questions and the observer serves as a coach should additionally be used. This method is called *coaching method* and allows to discover system elements that are not easy to understand by the children.

All sessions were recorded, in order to help and support the children at any time and not being busy writing a protocol. On top of this comes the fact, that even if the children would have been asked to write down their impressions or observations, it would have irritate and distract them too much and it is doubtful that they were be able to express themselves in a helpful way. The recorded tests were analysed afterwards and the used equipment is explained in subsection 5.2.2 (Lab and Material).

In the next subsection the tasks that were performed with primary school children are listed and explained.

5.1.3 Tasks

In order to examine the GUI Sugar, on the basis of defined criteria, a task list was developed which will be described in the following. First of all the tasks will be itemised and described, including assumptions or prospects what the children may do. Afterwards will be explained which facts of the heuristically comparison of Sugar and `Windows XP` affected the task list and in the end, these tasks are allocated to the criteria. This correlation could have been illustrated in the following task list, but is instead mentioned afterwards in the interest of readability.

It should be noted here, that some features of Sugar were explained to the children before the test, because it cannot be assumed that the children are able to find them out on their own. For example the Frame and the home button. Both are usable to return to the home screen. What exactly the children was told before the test is explained in more detail in subsection 5.3 (Procedure).

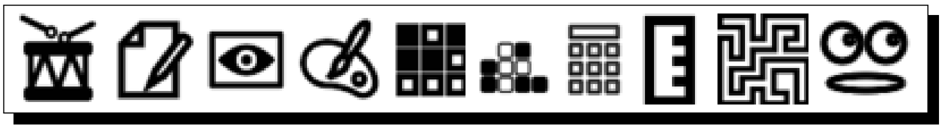

Figure 5.1: Sugar-Test activities

1. **Home screen icons:** The child should explain what kind of activities may be linked to the icons on the home screen as seen in figure 5.1. These icons are: TamTam (music jamming activity), Write (word processor), Record (still, video and audio capturing), Paint (paint activity), Memorize (memory game), Implode (logic game), Calculate (basic calculator), Ruler (graphical ruler), Maze (labyrinth game) and Speak (reader).
   Here the expressiveness and meaning of the symbols can be tested, from the children's point of view. Likely the children are able to identify most of the icons, but e.g. the symbols of the Implode and Memorize activity are maybe to hard to understand.

2. **Measurement activity:** The child should identify the Measurement activity – if not already happened in task 1 – and open it.
   It can be assumed that all the children know what a ruler is, but that does not mean that they are able to understand, that the ruler icon represents the Measurement activity. Perhaps it is the other way around: children are able to understand the symbol after they heard that there is a program to measure with.

3. **Measure:** The child should measure the length of a pencil – a pencil is provided, so they do not need to bring a pencil with them and start searching in their backpack – and memorise the length.

There is no doubt that the children are able to measure a pencil with a ruler, but due to the fact, that the measure activity provides more than one ruler on the screen, confusion could be the result.

4. **Back to the home screen:** The child should return to the home screen and remain the Measurement activity open. The reason why they should leave the activity open is the VNC activity that was already launched before the meeting (cf. 5.2.2 (Lab and Material)). When closing an activity the system returns only to the home screen if no other activity is open. If one or more activities are open, the system returns to the activity that was launched last. So that the children do not get irritated, they should only return to the home screen.
   This task should not be a problem, because the child was told how to return to the home screen before the test (cf. 5.3 (Procedure)).

5. **Calculate activity:** The calculator activity should be opened and to the memorised number from task 3 the age of the child should be added.
   Maybe some children are not sure what icon represents the calculator activity. Presumably entering the numbers will be easy for the children, but how they enter the numbers is not predictable: via the keyboard or the calculator buttons on the screen. Unsure is also if they are able to start the calculation, because the calculator has only an enter button, no German text or an equal sign.

6. **Copy to clipboard:** The result of the calculation should be copied to the clipboard.
   The clipboard was also explained before the test, but if the child needs help the clipboard feature will be explained again, and if still required additional support will be given.

7. **Close the activity:** The calculator activity should be closed.
   Here should be testes if the child is able to close the activity on her or his own. The activity can be closed with the octagon symbol that has a rectangle inside in the top right corner of the screen (cf. figure 5.4). If the children already gained some experience with `Windows XP` maybe they search for the `x`-button that is in every top right corner of a `Windows`-window to close it.
   As explained in task 4, when closing an activity the systems returns to the last launched activity, if it is still open. This means, SUGAR will not return here to the home screen, but to the Measurement activity and so the children need to return to the home screen manually.

8. **Speak activity:** The child should identify the speak activity, open it and is allowed to try some words or phrases out. The child has to find out on his or her own, what has to be done in order to let the XO speak the written text.

Afterwards the calculated number from the clipboard should be inserted and read by the XO.
If the children are able to the Speak activity can not be predicted, because the arrow next to the input field could be misinterpreted as the *start-reading* button. Instead the enter key has to be pushed. If the children need help to insert the number from the clipboard, it could be explaining that it is the opposite of task 6 and maybe some children are with this hint able to solve this task on their own.

9. **Close the activity:** The speak activity should be closed (cf. task 7).

10. **Record activity:** The activity for taking pictures should be identified and opened by the child (cf. task 2)

11. **Take a picture:** At least one picture should be taken by the child from themselves.
The shutter button is positioned directly under the video signal window, but it is not clear if the child are able to find it without help. Nevertheless it can assumed, when they identified the trigger, that the children will take more than one photo.

12. **Copy to clipboard 2:** The child should copy at least one picture to the clipboard.
Here should be identified if child is able to repeat the proceeding of task 6. And additionally the child should find out that it is not possible to copy a picture to the clipboard like the number in task 6. A right-click has to be performed on the respective picture and the copy to clipboard option chosen.

13. **Close the activity:** The record activity should be closed.
The difficulty is that there is no close button like in task 7 or 9. The record activity has to be closed with the accordant option in the context menu of the activity symbol in the upper part of the Frame. It can be assumed that they need help.

14. **Write activity:** The child should identify the Write activity and open it (cf. task 2)

15. **Insert from clipboard:** Both the number and the picture should be embedded from the clipboard in the document.
If the child was able to archive and understood task 8, she or he should also be able to perform this task.

16. **Close the activity:** The activity should be closed (cf. task 13).

17. **Labyrinth:** The child should find the labyrinth game and is allowed to play some turns on his or her own.

The labyrinth activity can be played with up to three persons, that means that there are three different control elements at hand. The game buttons on the left and on the right side of the screen and the arrow buttons on the keyboard. May be some children will first try to control the game with the mouse.

18. Labyrinth - Multiplayer: Child and coach should play some turns together. A little match between the child and the coach should close the session, so that the child can in any event left the test with a positive feeling.

All the tasks were tested before the real test sessions with an uninvolved person, who had never worked with the XO. This rehearsal took about 18 minutes and so it was assumed that 30 minutes are a sufficient time Frame for the real test sessions, with enough buffer. More about the procedure of the task is explained in 5.3.3 (Test Session).

In relation to the comparison of `Windows XP` and SUGAR in chapter 4 (Sugar vs. Windows) the following facts influenced the compilation of the task list:

Zoom metaphor: The zoom metaphor is partially tested, because the children have to return to the home screen several times by using one of the zoom buttons (in the Frame or on the keyboard).

The Frame: The Frame is also include in the test, because closing activities and copying elements to the clipboard is only possible within the Frame.

The Sugar-ring: The children should identify several elements on the home screen, which are part of the SUGAR-ring.

Nomenclature: Sometimes written language is used in SUGAR, for example in task 12, where a picture should be copied to the clipboard and context menu information has to be understood.

Generally speaking all six criterion can be applied to all tasks. A description at full length would repeat to much of the previous text. Instead by means of some specific examples this relation should be clarified in the following.

Several tasks allow to determine the systems' level of *consistency*: open and close activities (e.g. task 2, 5, 7 and 9) and the usage of the clipboard functionality (e.g. task 9 and 12).

Several features will not be explained to the children (e.g. how the XO is able to speak written text in task 8 or how to close an activity when no close button is at hand) and they need to find out on their own. The *self-descriptiveness* ability of SUGAR will tell if this is possible.

If the system respectively its elements are *conform with the users expectations* is explicitly tested e.g. in task 1, by asking the children what activities may be represented by the icons on the home screen.

If SUGAR provides miscellaneous *ways of operating* can also be tested by closing an activity, like task 9 compared to task 13. Another example is the labyrinth game, where more than one controller can be used to navigate to the exit.

The two last criterion *satisfaction* and *curiosity* cannot directly be related to one or more specific tasks, but rather result from the whole test session.

In the next section the target group, the search for a cooperative school and the equipment that was used during the test is explained.

5.2 Target Group and Setup

The choice of an appropriate target group is primary based on the XOs original target group: children. And since children cover a wide age spectrum, the target group is furthermore adjusted to their ability of taking part in an usability test. As already mentioned in subsection 3.4.4 (Usability Testing with Children) Hanna et al describe in their *Guidelines for usability testing with children* three different age ranges: preschool-aged children (2 to 5 years), primary-school-aged children (6 to 10 years), and middle-school-aged children (11 to 14 years). By reason that the children should be able to perform several tasks during the tests, with not that much computational experience, the appropriate target group are children from an elementary school between 6 to 10 years old. Younger children have difficulties to express their thoughts and may have problems when they are directed to perform tasks instead of self-exploration. Older children on the other side are able to criticise a computer system, but may have to much computer experience. Hanna et al. explain that due to their school experience, children between 6 and 10 years are able to sit at a task, follow directions, will answer questions and are willed to try out new things [Hanna et al., 1997, p.10]. But they can also be shy or inarticulate if they are ask to talk about the computer, what has to be considered during the meetings. As mentioned in 3.4.1 (Target group) five to six user should be tested to discover up to 80% of the usability problems [Nielsen, 1993] [Virzi, 1992]. Correspondingly at least six children have to perform the SUGAR-test.

It cannot be avoided that children within this target group already gained computer experience. According to the results of the *Kim-Studie 2008* – Kinder und Medien –, 56% of the children between 6 and 9 use the computer once or several times a week and 18% use it every or almost every day [MPFS, 2009]. Most of the time, they use the computer to play games, alone or with others, but also to surf the internet and work for school. Based on the statistics from W3Schools [Network Solutions, 2009], it can then be assumed that they most likely are familiar with `Windows XP`, because it is the most widespread and used operating system (cf. 4 (Sugar vs. Windows)). These facts can be beneficial in two ways. First of all, they do not have to be taught how to use a computer and secondly it can be examined in what extend the existing computational knowledge can be transferred from one operating system to another one.

5.2.1 ÜberMIttagbetreuung

Six different primary schools in the area of Lüneburg were contacted and asked if they would be interested to help performing the tests. An information leaflet was prepared and handed out to the respective school secretary, so that the necessary information could be forwarded to the responsible person. This leaflet is presented in A (School leaflet). None of the schools was able to agree on a cooperation right away and two schools directly decline to help, because they already had enough own projects and their rigid curricular allowed no additional ones. Further investigations relating after-school care clubs lead to a successful result. The care club from the Heiligengeistschule in Lüneburg called *ÜMI* – ÜberMIttagbetreuung – was willed and highly motivated to support the test. The staff of the ÜMI takes care of about 30 pupils from 12:3oh till 16:ooh on schooldays and from 7.45h till 16:ooh in school holidays. They ÜMI-logo is displayed in figure 5.2.

Figure 5.2: ÜMI Logo

A group of seven children was selected by the ÜMI-staff with girls and boys. A list with the seven children that participated is presented in table 5.1 – the names of the children are anonymised.

Name	Age	Gender	Class	Experience (+/-/o)
Gesa	7	F	1	o
Max	8	M	2	+
Dennis	7	M	1	o
Karl	10	M	3	+
Tina	7	F	2	+
Sonja	7	F	2	o
Jens	8	M	2	o

Table 5.1: List of all participants of the Sugar test

Even though the children estimate their computer experience differently, during the tests most of them were nearly all at the same level. None of them has their own personal computer at home, they all use the computer of their parents. Only one child owns a special learning computer. Most of the time they play games, but also surf the internet, draw pictures or write some text.

After all meetings and tests where finished only two schools got in touch and explained that they are willed to help. The other two school never answered my request.

In the next subsection the equipment and the test room is described.

5.2.2 Lab and Material

As already mentioned in subchapter 3.4.2 (Usability Lab), two types of usability labs exist: static and mobile labs. Static usability labs are commonly fixed equipped installations in a test and a control room. Mobile labs consist of portable equipment for a test that takes place outside a lab, for example to test a system under real conditions. Equipment for audio and video recording could be used for both types, in order to analysis the data at a later time. Due to the fact that the children reside at the ÜMI from 12:3oh to 16:ooh at the latest, it was more reasonable to use a mobile lab for the Sugar-test. Inviting the children to a static lab would be more complex and require more coordination. Another important point is that the children are already familiar with the local environment and they do not need to acclimate to a new surrounding. A well-known environment is also helpful if the children are shy, which is possible for children in this age (cf. 5.2 (Target Group and Setup)). Thus it is easier for them to open up, when they feel comfortable.

A separated room was provided by the director of the care club, that was used for the meetings without being interrupted. It is located directly next to the main room of the care club and normally used for homework doing or resting. In a critical situation it would have been possible to get help by one of the personnel. The room was already equipped with a large table and some chairs, which are adjusted to the size of primary school children.

In addition to the exploration and the test meeting with each child, an introduction meeting took place with all seven children. All meetings were recorded with different equipment. The introduction and the exploration meeting were recorded with a standard digital video camera with an integrated microphone. The camera was positioned on a tripod next to the table, so that the participant and the laptop were recorded from the front. After the meetings the recorded footage was dubbed to a computer for later analysis.

During the test meetings different equipment was used. In addition to the XO, a second laptop was installed to record the sessions. A USB-webcam and an external microphone connected to the second laptop were used to record video and audio and were positioned in front of the participant. On the XO a VNC activity was launched that allows to transfer the display signal via network to another computer. A network router was installed and configured, so that this signal could be send to and by means of an installed VNC-viewer picked up by the second laptop. Therefore it was possible to record also the display of the XO. A 15-day trial-version of the usability test-recording software *Morae* helped to record the signals from camera, microphone and VNC in one program. With this software it was also possible to analyse the recorded tests afterwards. Also a USB-mouse and a mousepad was brought for the children who were not able to cope with touchpad of the XO. The equipment was in all cases placed as undisturbing as possible and the children were informed that they are recorded. To be allowed to record the meetings all parents of the seven children received a paper with information about the project and test and additionally a letter of agreement. The letter describes that all the collected material will only be used for the project and not be available to other persons. This paper is displayed in B (Parents' consent). All letters of agreement returned signed by the parents.

In addition to the test recording some notes were taken during the meetings, which are also considered in the analysis section (5.4. Observations and Analysis). As in the comparison chapter (4. Sugar vs. Windows) the software version of the XO was stable build 767 (8.2.0) with the latest updates. More about the procedure of the meetings can be found in the following.

The procedure of all meetings is described in more detail in the next subsection.

5.3 Procedure

The sessions that were performed at the ÜMI with the seven children can be divided in three different types: *introduction meeting*, *exploration meeting* and *test session*. In the following the procedure of these meetings is explained.

5.3.1 Introduction Meeting

The first meeting took place with all seven children to get to know each other in the provided room at the ÜMI. This meeting had no time limit and lasted about 15 minutes. During this meeting the topic of the thesis and the test equipment were explained. The XO was shown shortly without opening or booting up. No project-related questions were asked by the children, but their interest was clearly noticeable, because they wanted to start using the laptop right away.

This meeting was recorded with a video camera, so that the children could get used to a camera. The video footage of this meeting was not that valuable, only the personal data of the children, like age, class and self-assessment of computer experience that were asked was relevant and is presented in table 5.1.

During this get-together it was made clear that not the children are being tested, like in a class exercise, but rather their help is needed and that they are the *experts* – which is true, because the XO was developed for children – who should test and review this computer respectively its GUI Sugar.

5.3.2 Exploration Meeting

With each child a 15 minutes exploration meeting took place. In this time each student was able to explore the XO on her or his own, without any task-fulfilment. The XO was placed in front of the children and they got the permission to play around with XO. During this meeting the behaviour and performance of the child were recorded with a video camera. For example how long it took to open the XO and what were their first impressions in relation to the XO. At first it was planned to record the display of the XO as well during this exploration time, but was deliberately omitted, because the meeting should only help the children to get used to the XO and a second camera could lead to confusion and irritation. Furthermore recording the video signal of the XO would have required that a specific activity had to be launched on the laptop before and this would not have allowed the children to start up the XO on their own. As already explained the children were informed that they were recorded and the camera influenced the sessions in no case. The exploration meetings took place on two consecutive days: four meetings on the first day and three on the second.

Based on didactic reduction, which is the reduction of complex data to its basic essentials [Hauptmeier, 1999]), not all Sugar-activities were presented on the home screen. Activities like the Terminal, Pippy, or the Distance Activity did not appear within the activity ring, because they may be too difficult for such a short test. Leaving out some of these activities lead to a better orientation, so that the child could focus on the essential activities. Additionally three of the activities that were part of the test meeting were also hidden during the exploration meeting, so that the children had a thoroughly original acquisition to these activities in the main session. The activities that were used during this meeting were the following: TamTam (music jamming activity), Write (word processor), Paint (paint activity), Memorize (memory game), Implode (logic game), Ruler (graphical ruler), and Speak (reader).

The settings how to activate the Frame were set to corners *and* edges (cf. 2.3.4

(The Frame)), because it was assumed that in this case it would be easier for the children to activate and become acquainted with the Frame.

5.3.3 Test Session

One week after the exploration meetings the test sessions took place. Four meetings were accomplished on the first day and three on the second. The length of the sessions was scheduled for about 30 minutes and the equipment was installed and checked on time. Unlike the exploration meeting not only the participant was recorded, but also the desktop of the XO respectively its video signal, as already explained in subsection 5.2.2 (Lab and Material). The XO was already powered up, so that the children were able to start immediately. Due to the fact, that eventually all children used the USB-mouse in the exploration meeting the mouse was connected to the XO right from the start. Three more activities were added to the home screen of the XO as also presented in figure 5.1: Calculate (basic calculator), Maze (labyrinth game) and Record (still, video and audio capturing).

Some features of the XO were explained at the beginning of test session: the Frame, the possibilities to get back to the home screen and the clipboard function. Returning to the home screen is possible with the accordant key on the keyboard or button in the upper part of the Frame. For this reason it was also explained how to activate the Frame respectively how to get the Frame in sight. In contrast to the exploration meeting, the settings were changed before the test session, so that the Frame activation was no longer possible with one of the four edges, only by moving the mouse to any corner on the screen. The Frame activation via the edges disturbed the children and is explained in more detail in 5.4.1 (Exploration Meeting - Observations). The clipboard function was also explained – mark the object that should be copied and drag it to the bottom left corner of the Frame –, in order to be able to copy data from one activity to another (cf. 5.1.3 (Tasks)). At the end of the session each child was asked what they liked and disliked about the GUI.

The observation that were made and the analysis of the data that was collected during the exploration and test meetings is presented in the following.

5.4 Observations and Analysis

In this section the observations from all meetings are listed and described, whereas the test meeting observations are revealed in relation to the tasks (cf. 5.1.3 (Tasks)) that the children had to perform and according to the testing methods that were used (cf. 5.1.2 (Testing Method)). Afterwards the appropriate data from these sessions is presented in relation to the defined criteria that was described in subsection 5.1.1 (Criteria). This segmentation between observations and analysis is necessary to obtain a connection between the recorded data and on one hand the task list and on the other hand the defined criteria. A mixture would certainly lead to a confusing text structure. It should be mentioned again that the meetings took place with seven children, because the number of children that for example reached a goal or had problems with a specific task or function of the GUI will occasionally be used in the following.

5.4.1 Exploration Meeting - Observations

At the beginning of the exploration meetings each child had to discover how to open the XO. None of the children was able to open the XO on their own, without asking for help. But eventually with some small hints all of them were able to unlock the top. Six children directly found the power button, because it looks the same as the button on the remote control of the TV at their home. The other child needed some time to detect how the XO can be booted up. During the boot sequence a schoolgirl explained the four colours that are shown when she start the computer at home: the four colours of the `Windows`-logo. The antennas of the XO were correctly identified as Wireless LAN receivers by two children and after the XO was opened one child recognised that the antennas lock the display when it is closed.

Only one of the children never used a touchpad before and five children used right away the USB-mouse that was offered. All had no problem connecting the mouse to one of the USB-slots. The other two switched to the mouse during the meeting due to handling-problems, because the cursor jumped uncontrollable around when using the touchpad. It was not possible to identify what caused this problem.

It did not take long and the children found out how to start activities from the home screen and none of them needed help. Most of the time, their prospects were satisfied, only two children mixed up the icons for the drawing- and writing activity. Their explanations what the symbol of the Speak activity could represent were very interesting: weird face, funny face and frog.

All children needed assistance returning to the home screen, so that another activity could be launched. By providing help they became acquainted with the Frame and the home screen key on the keyboard. In the following one girl was able to return to the home screen via the keyboard-button without further support. Three of them wanted to return to a previous and still launched activity, but instead of continue the activity – they did not noticed the activity icon in the top part of the Frame – they launched another instance of it from the home screen.

Three children opened the Tam-Tam activity and were very amused while creating some music with the different rhythms and selectable sounds. None of them needed help to figure out how to enable one of the rhythms, but it is assumed that this happened by accident or just by clicking around. One child was familiar with the scroll-wheel of the mouse and used the scroll-wheel to navigate down to more sound-buttons.

The Speak activity is indeed able to read written text in several languages, but the spoken welcome message “please type something” is only available in English, but pronounced in German. But none of the four children that used this activity was irritated by this and tried to figure out what the *face* is able to do, which was achieved only by one child. The enter key on the keyboard has to be pushed, therefore the XO reads the written text. The other three children thought the arrow next to the field needed to be pushed, but instead nothing happened. The arrows shows only all previous entered text from one session. Once the Speak activity was not executable due too less free-memory. Unfortunately this was detected not until after the sessions were finished.

The Draw activty was launched by four children during the exploration meeting. All of them were able to start drawing right away, with the preselected pen-tool. Three children wanted to change the draw-colour, which is possible with the colour-chooser in the top-left corner. Only one girl was able to find this chooser, but no one was able to do change to another colour without support. The colour has to be selected in two steps. First the main-colour has to be chosen from a colour-ring and afterwards the brightness of the colour has to be selected. These steps were too complicate without any further instructions. One girl selected the erase-function on her own and deleted some of the stroke in the picture.

The Frame which was adjusted to appear when reaching one side or corner of the screen, was very annoying for the children – in contrast to the assumption explained in 5.3.2 (Exploration Meeting). Every-time the Frame appeared, which happened mostly unintentional, the children were irritated and one girl even reacted a little bit angry and started to swear. Also the context menu, that appears when hovering for a moment over an icon, providing some settings or features, irritated at least five children more than it was supportive. Only two children used the context menu of the home screen-icons to identify the activity, because the name of the activity is displayed on top of the menu.

Five children wanted to close some of the activities but were not able to find out how. All of them were searching for the **x**-button in the top-right corner, which they know from the computer they use at home. It can be assumed that they were searching for the close button each `Windows`-window has in the top-right corner.

Figure 5.3: SUGAR warning symbols

Some of the children asked at the end of the test session if they should shut-down the XO. The method how to shut down the XO respectively which key or button has to be pushed to do so, was not found by any of the them. SUGAR does not come with a specific shut down-button placed somewhere in the direct visual field. Instead the shut down option is part of the context menu of the user icon in the centre of the home screen. But as already mentioned the children had some problems with the context menus. One child started to explain the symbols that appear during the shut down sequence without any mistake. The symbols are warning notices what should be avoided while handling the XO (cf. figure 5.3). For example: don't step on the XO, don't put it in water and don't stumble upon cord while plugged-in for charging.

The exploration meetings were totally underestimated. The exploration meeting showed, that the children are more familiar with a computer then expected. Unfortunately it emerged too late, that some of the children also have computer lessons in school. This is maybe cause by the fact that the students were tested in an after-school care club and not directly in school. Consequentially the task list, which was shorter in the first place, was expanded, so that the exercises are not too boring and meet the skills of the children.

5.4.2 Test Session - Observations

Before each session started some features of the XO were explained as already mentioned in 5.3.3 (Test Session), even though some of them were already used during the exploration meetings. Most of the children seemed to understand how to activate the Frame for using the accordant button to get back to the home screen or close an activity. But most of the children reacted very uncertain and partially irritated in relation to the clipboard function. It was not clear if they really understood this feature at that time. In the following the observations of the test session are pictured as to the task list the children had to perform (cf. 5.1.3 (Tasks)).

1. **Home screen icons:** Most of the icons were identified by the children. Some of them were already used during the exploration meeting. Very interesting were

the answers about the Ruler activity icon, as assumed. Three children only said “E”, because they thought the ruler represents the big letter “E”. Again some children identified the Speak activity as a frog, as in the exploration meeting. Some children were unsure about the icons of the memory and the Implode activity. They were often identified as an Calculate activity or the Tic Tac Toe game. And on the other side, the Calculate activity was identified as a game. The record activity – represents by an eye – was identified by all children as an activity for “seeing”, which in fact is correct, because the activity allows to record video and images.

Not really helpful was the context menu that appears when resting the mouse-cursor for a moment over an icon, because the name of the icon also appears within the menu. It was not assumed that the children use the mouse to point over an icon while describing them. And so some of the children read the name of the activity without thinking what it might be. As soon as this was detected the children were asked to identify the icons without using the mouse.

2. **Measurement activity:** All children were able to find the Measurement activity, even the ones that were confused by the “E”. They were directly able to find the activity after they were asked to measure something, accordant to the assumption in 5.1.3 (Tasks). One child was irritated by the different rulers that are distributed over the screen. One child activated this activity with a double-click and was not aware that a second instance of the program was launched. This happened with some other children and activities as well during the sessions.

3. **Measure:** All children were able to measure the length of the pen correctly and on their own.

4. **Back to the home screen:** One child returned to the home screen via the button on the keyboard and four via the icon within the Frame. Two children closed the activity with the icon on the top right corner of the Frame, thus they were not asked to do so. But this showed that for these two children the close button within the Frame was clear and understood.

5. **Calculate activity:** Only one child needed help to find the calculator. Five children entered the arithmetic problem by using the mouse and the other child used the buttons of the keyboard. Finding the button to get the result was not that easy. Only 4 children were able to do so without any help. One child wanted to use the equal-button on the keyboard, but was not aware that the shift-key has to be pushed at the same time. Unfortunately without success.

6. **Copy to clipboard:** One child used the copy-function of the edit-menu to copy the number to the clipboard, though it was explained in another way. From

the other six, only one was able to mark the number on her own with the mouse, because she already learned that in a computer-course. The rest of the children were not able to mark the number on there own and needed help. Dragging the number to the bottom left corner to copy it to the clipboard was performed by the six student only with support.

7. **Close the activity:** Three children were still searching for the **x**-button as in the exploration meeting, although they were told that there is no such symbol available. It can be assumed that they were searching for the `Windows-x`-button, because they searched all in the upper right corner of the Frame for the close-function, as in the exploration meetings. After activating the Frame, they identified the user symbol that is located in the upper right area of the Frame as the close button. But nevertheless the others were able to close the activity without any further help with the actual button within the Frame (cf. 5.4). Due to the fact that the Ruler activity was still launched – at least the 5 children that did not close that activity – the system did not return automatically to the SUGAR-ring but to the rulers. First the children were confused what to do next, but after explaining them why this activity is still opened and they have to return to the home screen for launching another program, they went ahead.

8. **Speak activity:** All launched the activity without problems. Before inserting the number from the clipboard, the three children that did not used Speak during the exploration meeting, were allowed to enter any kind of text to get used to the functionality and experienced the same *problem* as three of the other children: the enter key on the keyboard has to be pushed and not the arrow next to the text area.
   Inserting the number from the clipboard was done by all children with some help. They all understood that it is the opposite of copying something and eventually all of them heard the activity speak out their number.

9. **Close the activity:** All children were able to close the activity and return to the home screen on their own.

10. **Record activity:** The record activity was launched by all seven children without any help.

11. **Take a picture:** Only one child needed help taking a picture. But all of them took more than one picture as predicted in 5.1.3 (Tasks). One child referred to a mirror while exploring this activity.

12. **Copy to clipboard 2:** All children wanted to copy one of the pictures to the clipboard the same way they already did in task 6 and found out that it is not possible. They were told how to copy a picture to the clipboard – activate the

context menu of the picture and choose the 'copy to clipboard'-function – and eventually able to do so with more or less support.

13. **Close the activity:** In contrast to the other tasks where activities had to be closed, this activity has no close button in the upper right part of the Frame and all children were irritated and had no idea what to do next. Only with help they were able to close the activity and return to the home screen.

14. **Write activity:** No problems occurred while identifying and launching the Write activity.

15. **Insert from clipboard:** Before performing the actual task, all children played around with the activity and inserted different kind of text. Two children needed help to insert the number and the picture into the document. One of the other children even created an profile of herself and include the number and her picture in it.

16. **Close the activity:** Only one child still needed help to close an activity without the close button in the Frame .

17. **Labyrinth:** All children were able to launch the labyrinth, but only one child was able to use the arrow-keys on the keyboard to steer though the game. The others tried to use the mouse and needed help to find out how to control the game. While moving through the labyrinth the user icon moves automatically to the next turnoff. Four children identified this feature after they were asked.

18. **Labyrinth - Multiplayer:** All children were amazed that it is possible to play this game with up to three people with the controls on the left and right side of the screen.

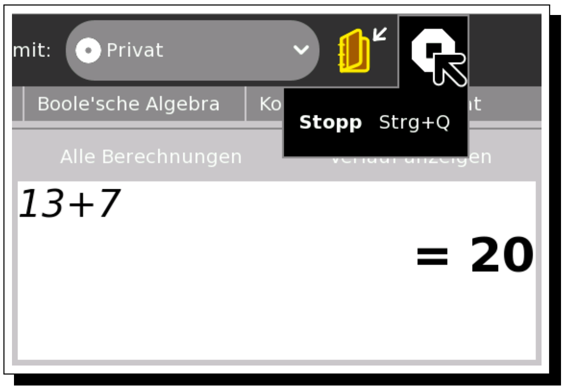

Figure 5.4: SUGAR close button

At the end of the session each child was asked what they liked and disliked about the GUI and the laptop itself. All of them said that they really liked the laptop and its design and for most of the children the laptop looks funny. In contrast to their willingness to accomplish the assigned tasks, they were partially insecure to express what they liked and disliked – this may be caused by shyness. Nevertheless everything was done, therefore the children feel as comfortable as possible. Nearly all of them mentioned games like the labyrinth as their favourite activity. Some others were also named: Tam Tam, Record and Speak. The only negative comments about the laptop was that it is slow, that the cursors sometimes jumps around and that the Frame is annoying.

All the test sessions were conducted without any insuperable problems. The most difficult situation for the children was the insertion of the clipboard-number into the Speak activity. This situation was totally underestimated in relation to its complexity. All the children were highly motivated and willed to perform the tasks, that they were asked to. But no child was able to perform all the tasks without support.

The average time the children needed to perform the tests was about 27 minutes. The fastest needed only 20 minutes and the slowest about 32 minutes to finish the test.

5.4.3 Analysis

Within this subsection the collected data from the exploration and the test meeting is presented according to the defined criteria in 5.1.1 (Criteria).

Consistency

The system itself – except the activities – is relatively consistent. For instance within all views the icons in the Frame are at the same position and the written text is mostly available in German. But if the activities are considered as well, the systems consistency leaves a lot to be desired. Most noticeable was the method of closing activities. For instance the Measurement activity and the calculator have an icon in the upper right corner to close the program. The Record or Write activity do not have such a button and if the user is not aware of the context menu functionality of the GUI, it is very hard to figure out how to close an activity.

Another point is the menu bar, that some activities have and some not. And the bars are not necessarily structured the same way if presents. Some menu bars have an entry like *edit* with functions for copying and pasting, as known from other operating systems. But others does not have this kind of entry. Such an inconsistent structure of menu bars could lead to confusion and irritation and the user need to get comfortable with these differences.

Not all activities respectively elements of activities are translated into German. For instance the enter button in the calculator. Only 4 out of 7 children found out, that this is the button to start the calculation.

Only a consistent structure of the interface and its programs – in this case activities – where the user is able to find elements and functions as easy and fast as possible, is able to support the user as effective as possible. “Users should not have to wonder whether different words, situations, or actions mean the same thing” [Nielsen, 2005].

The inconsistency of SUGAR is probably the result of the fact that many different developers design the activities. The activities should be tested – for instance by OLPC – according to *consistence-compartibility* before they can be installed on the XOs.

Self-Descriptiveness

Not all areas of the SUGAR-GUI were tested, for example the network capability was not part of the sessions. But the areas and activities that were tested left some of the children from time to time in the dark in relation to its self-descriptiveness capability. Already mentioned above: closing an activity. When there was no close-button – some children seemed to search for the `x`-button as in a `Windows`-based interface for closing a program window – all children had problem to figure out how to terminate the activity.

Other examples of unclear situations were:

Write: All children were able to write some text, but had problems to get to a new line. But to be fair, this may also happen if they would use another GUI.

Speak: How to start the *frog* talking or how to erase the text in the text area, was not accomplishable by all children. Nearly all were uncertain how to proceed and asked for help.

Clipboard: It was not obvious for any of the children how to copy any kind of data to the clipboard, although it was explained to them. It is questionable if anybody would be able to use this feature without being introduced.

Shut down the XO: As already explained some children asked to shut down the XO, but were not able to do so. There is no explicit button someplace in the GUI that provides this function.

Context menu: At some point the children had to use the context menu-function: task 12 (copying a picture to the clipboard) and 13 (closing an activity that does not provide an own close button). Though the text within the context menus is translated into German, some children were unsure what to do and needed some time to identify the correct procedure.

Positive examples should also be mentioned:

Sugar-Ring: All the children understood the functionality of the SUGAR-ring. All of them were aware that the icons represents programs or games that can be launched and used for different purposes. This may be the result of their already gained computer experience.

Games: The games that the children played (Maze, Implode, Memorize) were eventually understood by all children. For example the control of Maze was firstly tested with the mouse, but all concluded with the arrow-keys on the keyboard. Very impressive was one girl who was able to figure out and finish Memorize in a very short time.

Conformity with user expectations

As mentioned before, three children identified the user symbol of the XO that is located in the Frame by mistake as a close button. Another expectation has to be mentioned concerning the closure of activities. All of the children were not aware that closing an activity does not necessarily lead to the home screen. They were confused that the last launched activity is instead in sight (cf. 5.4.2 (Test Session - Observations)). The icons of the launched activities in the middle section of the upper Frame were not clear to the children, or they did not even noticed them.

Within the Speak activity all children misinterpreted the arrow next to the text area as the function that lets the *frog* read the written text. Instead of that, the arrow shows only the history of entered text of the actual session. Some of the children accidentally clicked with the mouse in another area of the screen. Consequently the text area was no longer *selected*. In this case it would be better – because the only function of Speak is inserting text and let the *frog* read this text out loud – that wherever the cursor might be, the text area should always be activated.

Ways of operating

Within Sugar are several operations possible by more than one way. Closing an activity is one of them. Both the close-icon as well as the accordant context menu entry can be used. Another fact are different activation possibilities of the Frame and the views. The mouse can be used to get the Frame in sight, but also the accordant button on the keyboard. Changing the views is possible via the accordant button on the keyboard, but also with the four view-icons in the upper part of the Frame. This was explained shortly to the children during the exploration meeting or at least before the test meeting, but all of them seemed to be overcharged and not sure which alternative they should choose. The meetings have shown that this irritated the children more than it was helpful. Sometimes they were unsure which function they should use. Maybe this distraction would disappear when they work with the system for a longer time, but during the exploration- and testing-phase it was not supportive.

As already mentioned in 5.1.1 (Criteria) too many ways of operating may confuse the user instead of provide any kind of support. In relation to the Frame- and view-button within the Frame and on the keyboard this might be justifiable, because one user prefers to use the mouse for such activities and another one the keyboard. A combination of both possibilities may be supportive if applied correctly. And the irritation of the children that was observed may disappear when working with the system for a longer period.

Satisfaction

It was already mentioned that all of the children really liked the laptop, especially its colour and the small antenna-ears. But there were also many situation were the children were very satisfied with the results and outcome they produced. For instance all of them were very happy after they found out how the XO is able to talk to them with the Speak activity, after they inserted their picture into the text document or when successfully finished one of the games.

But there were also some negative effects that were observed during the sessions. For example the processing power of the XO. At the beginning this was not a problem for the children, but the more activities they launched, the more bored they got while waiting for the laptop to be ready. Some of them even mentioned that their computer at home is much faster. In this context should be mentioned that after two children launched the Speak activity, it was not able to talk. This problem was solved fast, because too much memory was allocated to other opened activities and after they were closed everything was back to normal. It happened also two times that the Write activity crashed after the children solved task 15 and played around with the picture by changing the position and its size.

As described in the field reports and studies in 2.4 (XO-Experience) the children also experienced problems with the cursor. All of them used eventually an USB-mouse, because the touchpad was too inaccurate from time to time. But even while using the mouse two children experienced a situation, were the cursor jumped around for a couple of seconds.

During the exploration meeting the children considered the Frame as *annoying*, which is particularly the result of the settings. After the Frame-settings were changed only to the corners for the test meeting, the rate of non-intentionally activation of the Frame was lesser.

Another point was the context menu-function of SUGAR for some of the children. They were partially bothered and one girl even groaned every time the menu appears, because she did not want it to.

Some of the above mentioned are may caused by the hardware of the laptop, but such things should also be considered, because they appear while working with the

GUI and a child probably makes no difference between the hard- and the software when working with a computer.

Curiosity

After one boy accomplished the task with the Speak activity and took a picture of himself, he wanted to find out if the *frog* is also able to speak his picture. The system did not display an error-message or -sound, so it had to be explained that it is probably not possible to insert a picture in the text field. Unquestionable an interesting idea, but sadly he was disappointed, because his idea did not worked out.

Another child – mentioned already in 5.4.2 (Test Session - Observations) – wanted to create a profile of herself during task 15. Besides some personal data, she inserted also the calculated number and a picture of herself.

The presented results and its analysis will be concluded and commented in the next chapter.

6 Conclusion

USING TECHNOLOGY WISELY: THE KEYS TO SUCCESS IN SCHOOLS

Title of a book from Harold Wenglinsky [Wenglinsky, 2005]

In the last chapter of the thesis the results of the comparison of SUGAR and Windows XP and as well as from the SUGAR-test with primary school children will be concluded. The structure is oriented to the order of the previous chapters of the thesis and the questions that were raised so far will be considered. The report will be completed with a personal appraisal and outlook.

6.1 Resumé

At the end of chapter 2 (OLPC) several results of studies and reports of the XO were presented, that are related to the central question of the thesis (cf. 2.4.3 (Interim balance)). Three of the five results respectively observations were also made during the test with primary school children.

1. From time to time the cursor was inaccurate or jumped around and correspondingly difficult to handle by the children, especially when using the touchpad of the XO. A USB-mouse put some things right.
2. Nearly all children complained about the weak computing power of the XO.
3. None of the children was afraid about non translated text in some activities. Trial and error or asking for help was the procedure in such a case.

And on the other side, it did not occurred that the XO completely froze. Only the write activity had some problems – from jerky to the point of an activity crash – when inserting and modifying a picture. The opinion from Sharma that the XO "is very intuitive" [Sharma, 2008] can not be confirmed by the results of the performed tests. Nearly all children had problems with several functions and elements of SUGAR and were often only able to proceed after help was provided. An intuitive interface should be usable without previous introduction and explanation.

The assumption that the children may thread the XO as a toy and not as a learning tool can not been confirmed, because the tasks were too little and the time was too short to give a satisfactory and justified response. If the XO is able to prepare the children for the working world is personally commented in 6.2 (Appraisal & Outlook).

The heuristic comparison (cf. 4 (Sugar vs. Windows)) of SUGAR and Windows XP has shown that SUGAR may be more child-friendly that its opponent, because of its simpler structure – more icons than text – and the more colour- and playful design. During the test session with the primary school children it seemed that they really liked the design of the user interface. A counter-argument may be that the different interface elements of SUGAR (the four views, the Frame and the Journal) are too difficult to work with, but to be fair, most of them were eventually understood the more they were used.

The issues that were raised after the comparison of both systems on page 52 are also reviewed in 6.2 (Appraisal & Outlook), because the collected data is not sufficient enough for a distinct evaluation. Instead a personal personal assessment should be given.

As identified in 5.4 (Observations and Analysis) SUGAR has in relation to the defined criteria (cf. 5.1.1 (Criteria)) its advantages and disadvantages. The GUI itself is relatively consistent. The Frame provides across all views nearly the same layout and can be used well after the different areas and functions are understood. During the sessions none of the children was able to work out the functionality on her or his own. The user icon in the right area of the Frame was misinterpreted as a close button and most of the children totally ignored the rest of the Frame. Only one child spotted the icons of the launched activities in the upper Frame. And the clipboard function that was used for copying a number and a picture among different activities had to be explained to all the children more than once. After they were taught how to use these functions they performed better the more they use them.

Here should be mentioned again that during the exploration meeting the fade-in-settings of the Frame were set to the corners and edges, but the latter was very annoying for the children. Eventually all of them used an USB-mouse – the touchpad was not accurate enough and the children had trouble using it – and accidentally reached very often the edges of the screen, which activated the Frame. One girl even got a little bit angry and started to swear. This disturbance could be softened by changing the fade-in-setting only to the corners for the test session.

The several activities that are pre-installed or can be downloaded and installed later on the other side are not as consistent as they could be. Some of them have a close button and some not. Some provides a menu-bar, others do not and the ones that have a menu-bar integrated do not have a standardised structure of the menu. Additionally not all activities are translated completely. Especially the inconsistency of the close-button was difficult for the children. First they learned during the sessions that an activity can be closed with a button in the upper right corner of the window and later they are not able to find a way to close another activity, because no similar button is existent.

The self-descriptiveness of the interface is also patchy. The children needed indeed none or no great help to start activities or to play one of the games, but on the other side there were also some functions or activities the children were not able to understand. The most conspicuous was the Speak activity, were the arrow next to the text field was often misinterpreted as the trigger that lets the *face* talk.

The expectations of the children were also not meet constantly. One very interesting point was the confusion of the user symbol in the upper right corner of the

Frame with a close-button, which the children probably learned while working with the computer systems at home or in school – many GUIs have a small **x** in the top right corner of the program-windows for closing.

Some ways of operating that were already explained in the previous chapter (e.g. feature-activation with buttons on the screens or keys on the keyboard) can be used by the user in a positive and a negative way. If the user is constantly unsure which option she or he should take – which happened during the tests –, it can not be very supportive. But if the user is working with the mouse and needs to switch ofttimes between the different views, she or he can use the other hand that is resting on the accordant buttons to push them.

Some aspects – mostly hardware and not software issues – like insufficient computing power, a jumpy cursor and annoying events (Frame, context menu) led to an unsatisfactory atmosphere from now and then. But on the other hand the children were most of the time very pleased after they reached an expected goal and emphasised during and after the tests that they really liked the XO. The fact that two situations were observed were the children got curious – can the *frog* speak a picture and writing a personal profile (cf. 5.4.3 (Analysis)) – shows that SUGAR is able to motivate the children to get creative and try things out.

Nearly all of the above mentioned *negative* facts can be solved. A better processor and more memory would satisfy the user-needs regarding the processing power, but would increase the costs of the product. Missing close buttons, non-translated text, a jumpy cursor or inconsistent menu-bars can be improved, by adjusting the software. The problems some of the children had, how to use different functions of the interface may be prevented if they would be introduced to the system and its features or if the interface is reworked and improved.

Some suggestions of improvement should be given. A cooperation with teachers is useful for further investigation and would have been useful for this project as well. Unfortunately none of the contacted schools was in the position to help or reported back too late and so a collaboration with an after-school care club was the result (cf. 5.2 (Target Group and Setup)). Additionally, due to the fact that the children were all from different classes a high administration effort would have been necessary to collaborate with their teachers. Moreover it would have been interesting to record the keyboard of the XO for further analysis, which was not done during the tests. For instance, while some of the children tried to figure out what button needs to be pushed to display the result of the addition in the calculator-task (cf. 5.1.3 (Tasks)), several keys were tested and it was not possible to make notes of these attempts. This data could be useful to analyse what different keys the children associate to the wanted function.

An interesting question in this context is: Can such a system be supportive in the schooling of primary school children in Germany, after the discovered *bugs* are solved? But this issue can certainly not be answered with the presented results. More investigations would be advisable and necessary, to dig deeper into the subject matter, especially in learning matters. For further proceedings in this direction some recommendations should be given. Again, a cooperation with parents and teachers would be wise, so that on the one hand the childrens' computer experience is known and on the other hand the laptops can be tested under school conditions. Additionally a short introduction, where the children learn some of the basic functions of the GUI, can also be recommended, because the tests have shown that most of the irritations and misunderstandings that appeared during the tests could have been avoided if the children were familiar at least with the basics. Furthermore, a long-term study with more children would lead to more representative results, because the sample size would be bigger and maybe a correlation between computer experience of the children and the difficulties with the XO could be worked out. Moreover, according to Härtel, it seems that in Germany the prevailing view is that a too soon access to modern technique could obstruct writing skills, mental arithmetic or even the haptic handling with nature as he reported in the OLPC-issue of the trade magazine LOG IN [Härtel, 2009]. This may also be examined, before an application in German schools.

In the next section a personal appraisal and an outlook will be given.

6.2 Appraisal & Outlook

Some of the questions that were raised in 2.4.3 (Interim balance) and 4.4 (Summary) can not be answered satisfyingly with the presented results, but instead a personal estimation will be given.

Although the *Journal* of the SUGAR-GUI was not tested during the sessions at the ÜMI, it was nevertheless part of the comparison process as presented in 4.1.5 (Explorer vs. Journal). The question was raised, if such a file system is able to survive on the market, in this case for German school children. It is obvious that such a method supports the user in a way, where she or he is able to use the system without handling with file names and file locations. But the system provides no possibility in the event that the user – when getting older – wants to decide her- or himself where and with which name an object respectively file should be saved. An alternative could be a file system that supports both methods and the user is able to switch at any given time. Consequentially an interlocked functioning is required, so that automatically created files are also trackable and manageable manually.

According to the statistics from w3schools – `Windows` especially `Windows XP` is the most often used operating system respectively GUI [Network Solutions, 2009] – it can be assumed, that after the school education in Germany a large number of children will deal with this system in their later working environment. And in this context, it is interesting and appropriate to ask, to which extend is SUGAR able to prepare children for their later working environment? Would it be problematic for the children to learn on SUGAR and switch later to another interface? Or wouldn't it be better to start directly with a system that is prevalent worldwide? Answering these question satisfactory is very difficult, would require further research and go beyond the scope of the thesis. But due to the fact that SUGAR was developed to promote childrens' education through collaborative and creative activities, I personally think that the skills that could be acquired, like teamwork or creativity, may form a solid basis for the upcoming school and training career, so that such a switching process may not be that hard. Moreover, we need to adapt to new situations all the time, we have to relearn and rethink our workings and learning new things is still the foundation of education and training.

I personally think, that if the presented drawbacks of the SUGAR-interface would be fixed and children get a small introduction about the structure and features of the system, a lot of fun and creative output could be the result.

As mentioned before in 2.1 (OLPC) a delivery of the XO is planned with an adapted version of `Windows XP`, more precisely with both systems, SUGAR and `Windows XP` thanks to the additionally developed dual-boot capacity of the XO. According to an article of IT World Microsoft plans to offer their software package consisting of a version of `Windows XP`, Microsoft Office Home and Student 2007, Microsoft Math 3.0, Learning Essentials 2.0 for Microsoft Office and Windows Live Mail, for only $3 [Lawson, 2008]. Accordingly it will be possible in the future to switch between both systems in relation the aspired task. In this context, the statement of Negroponte mentioned on page 5, that SUGAR should have been developed as an application for an already existing operating system is very interesting. Although Scott Ananian from `laptop.org` pointed out that there are fundamental reasons [Ananian, 2008] why SUGAR was and is not supported by `Windows` – different design guidelines and the activity approach are just two examples – it would nevertheless be an attractive approach if SUGAR would be an independent application or package that is able to run on different operating systems. Furthermore one could create a version of SUGAR with different knowledge-layers, according to the performance level of the user. For example every higher level could have more text and less symbols and could be continued until the level of a *normal* operating system is reached.

While working on the thesis respectively working and examining the XO, its slowness was evident, as well as noticed by the children during the test. This observation contradicts the statement from the OLPC project, that the power of the XO is sufficient enough for the activities that are installed. If the speed of the XO would not be improved it is doubtful, that primary school children in Germany are willed to work with the laptop for a long term period.

Even though OLPC has still not reached the desired price of $100 for the XO-1, a new version of the laptop is in preparation. It is planned – according to Negroponte – to release the XO-2 in 2010 with several new features and a price of $75. The most significant innovation of the new version will be its display respectively its two displays. The XO-2 shall have a dual touch-display with the size of a book. One of the display can either serve as a keyboard and the other as the display, or both can be used as a display for several activities, for instance reading a book or playing a game like Ping-Pong with another person. Furthermore the energy consumption should be reduced to only one Watt. It remains to be seen if this new laptop, especially at that price, will be released.

Finally, a personal comment about the usage of the XO in primary schools in Germany should be given, even though it is not part of the key question. Based on the findings that were presented in 6.1 (Resumé) and that further investigation would make sense, it is clear that SUGAR is definitely not unhealthy. In my opinion the XO is ideal for younger students without any computer experience. It has a nice and colourful interface and with professional guidance, an introduction into the world of computers and digital media should be possible. The school is moreover able to adapt the activities of the XO to their needs, because of the FOSS. Due to the fact, that the extent of the SUGAR-interface is less than other GUIs e.g. `Windows XP`, it is also easier to introduce the system to teachers, therewith they can teach the students how to work with it. Whether it is really necessary that the system is running on the XO-1 laptop is doubtful. Because the tests have shown that after a short *discovery-phase*, where the children were amazed and delighted about the look and feel of the laptop, they concentrated more on the display, thus only on the activities, than on the laptop itself. Furthermore all of them used eventually the provided USB-mouse and not the touchpad. Since the SUGAR-interface is also available as a Live-CD and can be used on nearly all common laptop- or desktop-computers, it would be easier and cheaper to use the system on already existent computers.

References

[Alliance for Childhood, 2001] Alliance for Childhood (2001). Fool's gold: A critical look at computers in childhood. http://www.allianceforchildhood.net/projects/computers/. Last visited on 2nd October 2009.

[Ananian, 2008] Ananian, C. S. (2008). A technical assessment of porting "sugar" to windows. http://lists.laptop.org/pipermail/devel/2008-April/013242.html. Last visited on 2nd October 2009.

[Bailey, 1999] Bailey, B. (1999). Human factors international. http://www.humanfactors.com/downloads/may99.asp. Last visited on 2nd October 2009.

[Bevan et al., 1991] Bevan, N., Kirakowski, J., and Maissel, J. (1991). What is usability? In *Proceedings of the 4th International Conference on Human Computer Interaction*.

[Bigham, 2007] Bigham, J. P. (2007). Enabling blind web users to create accessible web content.

[Cable News Network, 2009] Cable News Network (2009). Fortune 500. http://money.cnn.com/magazines/fortune/fortune500_archive. Last visited on 2nd October 2009.

[Derndorfer et al., 2009] Derndorfer, C., Berry, B., and Vota, W. (2009). Olpc afghanistan's first day of school in jalalalbad. http://www.olpcnews.com/. Last visited on 2nd October 2009.

[DIN EN ISO 13407, 1999] DIN EN ISO 13407 (1999). Human-centred design processes for interactive systems.

[DIN EN ISO 6385, 2004] DIN EN ISO 6385 (2004). Ergonomic principles in the design of work systems.

[DIN EN ISO 9241, 1998] DIN EN ISO 9241 (1998). Ergonomic requirements for office work with visual display terminals (vdts) - part 11: Guidance on usability.

[Dumas and Redish, 1999] Dumas, J. S. and Redish, J. C. (1999). *A Practical Guide to Usability Testing*. Intellect Books.

[Eason, 1984] Eason, K. D. (1984). Towards the experimental study of usability. In *Behaviour and Information Technology*, volume 23, pages 133–143.

[Fildes, 2007] Fildes, J. (2007). Politics 'stifling $100 laptop'. http://news.bbc.co.uk/2/hi/technology/7094695.stm. Last visited on 2nd October 2009.

[Fried and Beiersmann, 2008] Fried, I. and Beiersmann, S. (2008). Bill Gates: Verbreitung des OLPC ist unbedeutend. silicon.de http://www.silicon.de/cio/strategie/0,39038989,39161420,00/bill+gates+verbreitung+des+olpc+ist+unbedeutend.htm. Last visited on 2nd October 2009.

[Gediga and Hamborg, 2002] Gediga, G. and Hamborg, K.-C. (2002). Evaluation in der software-ergonomie. *Zeitschrift für Psychologie*, 210:40–57.

[Hanna et al., 1997] Hanna, L., Risden, K., and Alexander, K. (1997). Guidelines for usability testing with children. *Interactions*, 4(5):9–14.

[Hanna et al., 1998] Hanna, L., Risden, K., Czerwinski, M., and Alexander, K. J. (1998). The role of usability research in designing children's computer products. *The design of children's technology*, pages 3–26.

[Härtel, 2009] Härtel, H. (2009). Das OLPC-Pilotprojekt in Äthopien. *LOG IN*, 156:18–22.

[Hauptmeier, 1999] Hauptmeier, G. (1999). Didaktische reduktion bzw. pädagogische transformation. *Wörterbuch Berufs- und Wirtschaftspädagogik*, pages 175 – 178.

[Heise, 2007] Heise (2007). 15.000 olpc-laptops gehen nach alabama. http://www.heise.de/newsticker/15-000-OLPC-Laptops-gehen-nach-Alabama--/meldung/100229. Last visited on 2nd October 2009.

[Hom, 2003] Hom, J. (2003). The usability methods toolbox. http://jthom.best.vwh.net/usability/. Last visited on 2nd October 2009.

[Hourcade et al., 2008a] Hourcade, J. P., Beitler, D., Cormenzana, F., and Flores, P. (2008a). Early olpc experiences in a rural uruguayan school. In *CHI 2008 Workshop on HCI for Community and International Development*.

[Hourcade et al., 2008b] Hourcade, J. P., Beitler, D., Cormenzana, F., and Flores, P. (2008b). Reflections on a pilot olpc experience in uruguay. In *CHI 2008 Workshop on HCI for Community and International Development*.

[Jackson, 2009] Jackson, A. (2009). A laptop for every poor child. http://www.zdnetasia.com/insight/hardware/0,39043471,62056166,00.htm. Last visited on 2nd October 2009.

[Jastrzębowski, 1857] Jastrzębowski, W. (1857). *An outline of ergonomics, or the science of work based upon the truths drawn from the science of nature*. Nature and Industry. Reprinted by the Central Institute for Labour Protection (1997).

[Kafai and Resnick, 1996] Kafai, Y. B. and Resnick, M. (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World.* Lawrence Erlbaum Associates.

[Kay, 1990] Kay, A. (1990). User interface: A personal view.

[Lawson, 2008] Lawson, S. (2008). Olpc, microsoft to deliver dual-boot xo laptop. `http://www.itworld.com/olpc-microsoft-to-deliver-dual-boot-xo-laptop-080516`. Last visited on 2nd October 2009.

[Leeming, 2008] Leeming, D. (2008). Some feedback on challenges and impacts of olpc. `http://www.olpcnews.com/files/OLPC_Oceania_Impacts_and_Feedback.pdf`. Last visited on 2nd October 2009.

[Lenhart et al., 2007] Lenhart, A., Madden, M., Macgill, A. R., and Smith, A. (2007). Teens and social media - the use of social media gains a greater foothold in teen life as they embrace the conversational nature of interactive online media. `http://www.pewinternet.org`. Last visited on 2nd October 2009.

[Lowes and Luhr, 2008] Lowes, S. and Luhr, C. (2008). Evaluation of the teaching matters one laptop per child (xo) pilot at kappa iv. Institute for Learning Technologies Teachers College/Columbia University.

[LukeW Interface Designs, 2006] LukeW Interface Designs (2006). Interface design quotes. `http://www.lukew.com/resources/quotes.asp`. Last visited on 2nd October 2009.

[MaMaMedia Inc., 2009] MaMaMedia Inc. (2009). Professor Seymour Papert. `http://www.papert.org/`. Last visited 2nd October 2009.

[Martinazzo et al., 2008] Martinazzo, A. A. G., Patrício, N. S., Biazon, L. C., Ficheman, I. K., and Lopes, R. d. D. (2008). Testing the olpc drawing activity: an usability report. In *Eighth IEEE International Conference on Advanced Learning Technologies.*

[Mayhew, 1999] Mayhew, D. J. (1999). The usability engineering lifecycle. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 147–148, New York, NY, USA. ACM.

[MPFS, 2009] MPFS (2009). Medienpädagogischer Forschungsverbund Südwest – KIM-Studie 2008 - Basisuntersuchung zum Medienumgang 6- bis 13-Jähriger in Deutschland. `http://wwww.mpfs.de/fileadmin/KIM-pdf08/KIM08.pdf`. Last visited on 2nd October 2009.

[Negroponte, 2008] Negroponte, N. (2008). Mission statement of olpc. `http://lists.laptop.org/pipermail/sugar/2008-May/005795.html`. Last visited on 2nd October 2009.

[Negroponte, 1995] Negroponte, N. P. (1995). *Being Digital*. Vintage.

[Negroponte, 2005] Negroponte, N. P. (2005). The hundred dollar laptop – computing for developing nations. http://mitworld.mit.edu/video/313. Last visited on 2nd October 2009.

[Network Solutions, 2009] Network Solutions, L. (2009). Os platform statistics. http://www.w3schools.com. Last visited on 2nd October 2009.

[Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*. Boston: Academic Press.

[Nielsen, 2005] Nielsen, J. (2005). Ten usability heuristics. http://www.useit.com/papers/heuristic/heuristic_list.html. Last visited on 2nd October 2009.

[Nielsen and Molich, 1990] Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proc. ACM CHI'90 Conf.*, pages 249–256.

[OLPC, 2009] OLPC (2009). One laptop per child (olpc), a low-cost, connected laptop for the world's children's education. http://www.laptop.org. Last visited on 2nd October 2009.

[Papert, 1981] Papert, S. (1981). Computers and computer cultures. *Creative Computing*.

[Papert, 1991] Papert, S. (1991). Constructionism: A new opportunity for elementary science education. Award Abstract #8751190. Submission for a National Science Foundation.

[Perton, 2006] Perton, M. (2006). Gates proposes cellphones as alternative to olpc. http://www.engadget.com/2006/01/30/gates-proposes-cellphones-as-alternative-to-olpc/. Last visited on 2nd October 2009.

[pressetext, 2006] pressetext (2006). 100-dollar-laptop gibt es nicht unter 150 dollar. http://pressetext.at/news/061118010/100-dollar-laptop-gibt-es-nicht-unter-150-dollar/. Last visited on 2nd October 2009.

[Sarodnick and Brau, 2006] Sarodnick, F. and Brau, H. (2006). *Methoden der Usability Evaluation*. Huber.

[Schelhowe, 1997] Schelhowe, H. (1997). *Das Medium aus der Maschine: zur Metamorphose des Computers*. Campus Verlag.

[Senajova, 2008] Senajova, S. (2008). Olpc afghanistan starting! http://lists.sugarlabs.org/archive/iaep/2008-October/002109.html. Last visited on 2nd October 2009.

[Sharma, 2008] Sharma, U. (2008). Formative evaluation of olpc project nepal: A summary. http://blog.olenepal.org/index.php/archives/321. Last visited on 2nd October 2009.

[Shneiderman, 1992] Shneiderman, B. (1992). *Designing the user interface (2nd ed.): strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Smith and Mosier, 1986] Smith, S. L. and Mosier, J. N. (1986). *Guidelines For Designing User Interface Software*. The MITRE Corporation.

[Somervell et al., 2003] Somervell, J., Wahid, S., and McCrickard, S. (2003). Usability heuristics for large screen information exhibits. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'03)*.

[Spillers, 2007] Spillers, F. (2007). Usability from wwii to the present- the historical origins of usability testing. http://www.usabilitytestingcentral.com/2007/02/the_history_of_.html. Last visited 2nd October 2009.

[Spool, 2007] Spool, J. (2007). The difference between usability and user experience. http://www.uie.com/brainsparks/2007/03/16/the-difference-between-usability-and-user-experience/. Last visited on 2nd October 2009.

[Thornburg, 2008] Thornburg, D. D. (2008). Is there enough sugar in your educational technology diet?

[Virzi, 1992] Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Hum. Factors*, 34(4):457–468.

[von Glasersfeld, 1990] von Glasersfeld, E. (1990). An exposition of constructivism: Why some like it radical. In *Monographs of the Journal for Research in Mathematics Education*.

[Vota, 2007] Vota, W. (2007). Olpc xo btest-1 units span a kilometer in wifi connectivity. http://www.olpcnews.com. Last visited on 2nd October 2009.

[Wedekind and Koerber, 2009] Wedekind, J. and Koerber, B. (2009). Laptops statt schulranzen. *LOG IN*, 156:3.

[Wenglinsky, 2005] Wenglinsky, H. (2005). *Using technology wisely: The keys to success in schools*. Teachers College Press.

A School leaflet

Hallo,

meine Name ist Sven Bergmann, ich studiere Digitale Medien an der Universität Bremen und schreibe gerade meine Abschlussarbeit über das Hilfsprojekt *One Laptop Per Child*. Im Rahmen dieses Projektes wurde ein Laptop mit einer bisher einzigartigen Benutzeroberfläche entwickelt. Dieser soll hauptsächlich an Kinder in Entwicklungsländern verteilt werden, um ihnen die Möglichkeiten zu bieten mit Hilfe des Internets zu lernen und mit anderen Kindern zu kommunizieren und zusammenzuarbeiten.

In meiner Abschlussarbeit möchte ich diesen Laptop mit sieben Grundschulkindern testen, um die Vor- und Nachteile dieses Laptops herauszufinden. Ihre Schule wurde mir als sehr medienorientiert empfohlen.

Ich würde diesen Test gerne in drei Phasen durchführen: 1. Ein kurzes Treffen mit allen sieben Kindern zum Kennenlernen und Erklären, um was es in diesem Test geht (Es wird der Computer getestet und nicht die Kinder). 2. Kurze Einzeltreffen mit jedem Kind von ca. 15 Minuten, in denen jedes Kind den Laptop kennenlernen kann. Und 3. Eine knapp 30 minütige Testphase mit jedem Kind, in der diverse Aufgabenstellungen durchgeführt werden sollen (z.B. das Malen eines Bildes).

Damit ich während der Tests den Kindern zur Seite stehen kann, möchte ich die Treffen gerne auf Video aufzeichnen, um im Nachhinein eine Auswertung vorzunehmen. Ich würde gerne mit Kindern arbeiten, die noch nicht viel Erfahrung im Umgang mit Computern haben. Da ich nicht sicher bin inwieweit sich 6-10 Jährige bereits auskennen, bin ich auf ihre Empfehlung angewiesen.

Falls Sie Lust haben mich in meinem Projekt zu unterstützen, würde ich mich über ein Treffen mit Ihnen freuen, um den weiteren Verlauf zu besprechen.

Mit freundlichen Grüßen
Sven Bergmann

Grenzstrasse 3
21337 Lüneburg
sven-bergmann@gmx.net
04131/7207828

Figure A.1: School leaflet

B Parents' consent

digitalemedien Universität Bremen

Hallo,

meine Name ist Sven Bergmann, ich studiere Digitale Medien an der Universität Bremen und schreibe gerade meine Abschlussarbeit über das Hilfsprojekt *One Laptop Per Child*. Im Rahmen dieses Projektes wurde ein Laptop mit einer bisher einzigartigen Benutzeroberfläche entwickelt. Dieser soll hauptsächlich an Kinder in Entwicklungsländern verteilt werden, um ihnen die Möglichkeiten zu bieten mit Hilfe des Internets zu lernen und mit anderen Kindern zu kommunizieren und zusammenzuarbeiten. Auf *http://de.wikipedia.org/wiki/OLPC_XO-1* sind weitere Informationen zum Projekt und Laptop zu finden.

In meiner Abschlussarbeit möchte ich diesen Laptop mit sieben Grundschulkindern testen, um die Vor- und Nachteile dieses Laptops herauszufinden. Diesen Test würde ich gerne in drei Phasen durchführen: 1. Ein kurzes Treffen mit allen sieben Kindern zum Kennenlernen und Erklären um was es in diesem Test geht (Es wird der Computer getestet und nicht die Kinder). 2. Kurze Einzeltreffen mit jedem Kind von ca. 15 Minuten, in denen jedes Kind den Laptop kennenlernen kann. Und 3. Eine knapp 30 minütige Testphase mit jedem Kind, in der diverse Aufgabenstellungen durchgeführt werden sollen (z.B. das Malen eines Bildes). Anschliessend sollen die Kinder noch ein paar Fragen beantworten, wie ihnen der Laptop und der Test gefallen haben.

Damit ich während der Tests den Kindern zur Seite stehen kann, möchte ich die Treffen gerne auf Video aufzeichnen, um dann im Nachhinein eine Auswertung vorzunehmen. Die gesammelten Daten werden ausschliesslich für die Auswertung während meiner Projektarbeit verwendet und nicht anderweitig eingesetzt oder an Dritte weitergegeben. Alle in meiner Arbeit angegebenen Daten werden anonymisiert. Dafür benötige ich ihr Einverständnis und bitte Sie daher die unten anhägnede Erklärung zu unterschreiben.

Falls Sie Fragen zu meinem Projekt haben, können Sie mich gerne jederzeit Kontaktieren oder dies auf dem unteren Abschnitt vermerken.

Mit freundlichen Grüßen und herzlichen Dank
Sven Bergmann

Grenzstrasse 3 - 21337 Lüneburg - sven-bergmann@gmx.net - 04131/7207828

- -

☐ Ich bin damit einverstandenm dass mein Kind an oben genannten Test teilnimmt.

☐ Ich habe folgende Frage: ______________________

____

______________________ Kontakt: ______________________

Name des Kindes ______________________

Unterschrift, Datum ______________________

Figure B.1: Parents' consent

List of Figures

List of Tables

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Sven Bergmann

Lüneburg, 2nd October 2009