In search of contextual teaching of programming in a Tanzanian secondary school

Marcus Duveskog[1], Erkki Sutinen[2], Matti Tedre[3], Mikko Vesisenaho[4]

Abstract *- Teaching programming in non-Western surroundings reveals the cultural roots and dependencies of Computer Science. Both the concepts and the teaching methods of the discipline needed to be rethought in a teaching experiment carried out in Kidugala, Tanzania, among secondary school students. Following the idea of contextualized Computer Science, called ethnocomputing, we used a culturally relevant entry points to teach the basics of programming. HIV/AIDS was chosen as a topic of an Internet site, to be designed and implemented by novice programmers, using the Java language. Analyzed by action research, our experiences indicate a significant motivation among the students to learn programming skills, in order to be able to deal with a taboo-like topic in neutral platform - a computer. The experiment suggests that a culturally relevant entry point, combined with problem-based learning, could challenge novice programmers also in Western societies; a side-effect of studying Computer Science education in a less developed country.*

SCIENCE AND CULTURE

The foundations of Computer Science (CS) were laid in Europe and America, and the development of computers, computer networks, and information and communication technology (ICT) are all Western products. The interests of military and intelligence (from "Colossus" to the Cold War to "Carnivore" to nuclear reaction simulations), economics, and other sciences have guided the directions of the development of computers from the beginning.

Being Western products, the teaching material, literature, and problem solving methods of Computer Science are dominantly Western. This inevitably causes problems in Computer Science education (CSE) in non-Western cultures, for a considerable amount of understanding about Western system of knowledge (e.g. inference, quantification, comparison, classification, representation, and measuring is assumed [1]).

In the West, technology is often seen as something that is given from outside, *deus ex machina*, rather than as a product of the society. Societies embrace new technologies that the enclosed technological circles produce. However, the knowledge of computer scientists is inevitably detached from societal context, and their interests may rather be those of big corporations or pure science, than those of common people.

The Internet has been labeled as both a major cause and a major embodiment of inequality between developed and developing countries. However, the Internet is also a Western product, and its first users and developers have already shaped the Internet for latecomers, both in terms of content and technology [3]. We see that not mere increase in the use of technology, but a cultural contextualization of education and educational tools is necessary to empower the people from developing countries to participate in shaping this international network.

CS students occupy a special place in the development of the Internet and ICT in general. Compared to the citizens who are or become consumers of the Internet and preferably even its producers [3], Computer Science students that graduate are already producers of the ICT (not only the Internet) by definition. System designers, programmers, and computer engineers are automatically change agents in the communities that aspire to assimilate ICT.

If information and communication technologies are to reach their full potential in developing countries, it is imperative that the needs of the local societies work as the starting point of CSE. Instead of intentionally looking for problems that the Western Computer Science could solve, the focus of science has to be changed, adjusted to solve the real problems at hand in developing countries. Those problems and needs may be very different from the needs of Western countries, and adjusting education to address those needs may require us to view CS in an untraditional manner. Even though the technology would be foreign to the cultures of developing countries, the uses of technology are what matters.

People, institutions, and society at large transform technology – any technology – by appropriating it, by modifying it, and by experimenting with it [3]. Technological systems are socially produced, and social production is culturally informed. This leads us away from positivism, toward the theory of the social construction of

[1] Marcus Duveskog, University of Joensuu, Department of Computer Science, madu8689@student.uu.se
[2] Erkki Sutinen, University of Joensuu, Department of Computer Science, erkki.sutinen@cs.joensuu.fi
[3] Matti Tedre, University of Joensuu, Finland, mmeri@cs.joensuu.fi
[4] Mikko Vesisenaho, University of Joensuu, Department of Computer Science, mvaho@cs.joensuu.fi
The development research is funded by the Academy of Finland.

reality. Repetitive patterns of behavior generate customs that are first enforced by institutions and organizations, and finally adopted as something non-human [4].

Science is no exception. Throughout time, people have felt their current view on science is something greater than human. Scientific paradigms, how ever solid and eternal they seem, are nothing else than social constructs, since they are essentially the result of an agreement among a scientific community [5]. If we agree that a paradigm is the result of a consensus among scientists, we can argue that debates on knowledge claims are valid only within a restricted community in a certain era. If we can agree with this, we should leave room for other epistemologies than ours, and essentially, we should give culturally bound knowledge the chance to support the students' mental imagery in learning.

Our claim of ethnocomputing [6] suggests that the conceptual understanding of Computer Science differs between cultures as well. We defined computing as a combination of four levels: (1) the organized structures and models used to represent information (data structures); (2) the ways of manipulating the organized information (algorithms); (3) the mechanical and linguistic realizations of the former; and (4) the applications of all the three former levels. Different cultures may have different views on the concepts on each level, and this should be taken into account in CSE. Moreover, science must be able to adapt to the needs of the society, and likewise, the society must be able to respond to the new technologies and their uses. Otherwise science will only create new needs to those societies, and not solve anything.

The consequences of the inevitable changes that ICT will bring can be unpredictable. Does taking ICT into feudal economies allow (or force) them to skip certain phases, going straight into postmodern, knowledge economies that are extremely volatile to changes in global economy? It seems that the village economies will have to step from eroding village communities (young people leaving to the big cities once they finish school) into self-sustained communities with one access point (now in African countries usually the post office) where farms and souvenir industry may orient to addressing new markets and finding knowledge on how to optimize production methods.

CULTURALLY CONSIDERATE EDUCATION

Geert Hofstede has illustrated human mental programming as a three-level pyramid (see Figure 1), where *personality* is the unique set of mental programs that an individual does not share with other individuals; *human nature* is what all human beings have in common; and *culture* is the collective programming of the mind that derives from one's social environment [2]. We are interested in the influence of culture on Computer Science education. We see that cultural understanding works as a key to understanding how Computer Science should be taught in non-Western countries. However, rather than focusing on the level of individual differences, we still concentrate on the mass level.

Technical concepts of Computer Science are not inherited; they are *not* embedded in the universal human nature. They have been developed within a culture, and are learned in a community. Thus, they are specific to a group or category, e.g. Western scientific community, to some degree. Even though conceptual understanding of Computer Science shares more features *between* cultures than some other cultural aspects do, it is still a societal product. Instead of taking our categorizations and ideas to the developing countries, we would like to see CSE originating from the cultures, using the cultures' concepts, artifacts, and knowledge to contextualize the science.

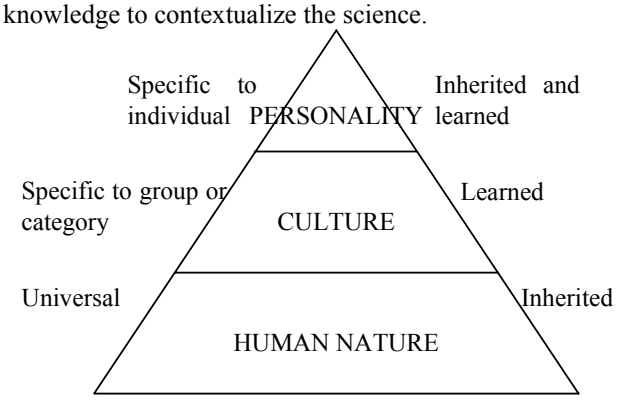

Figure 1: Three levels of uniqueness in human mental programming [2].

Our viewpoint is a social constructivist one in a sense that we believe that what we take to be objective knowledge and truth is a result of perspective [23]. Knowledge and truth are created, not discovered by the mind. In the constructivist paradigm, the realities are socially and experientially based [23]. The social constructivists claim that any form of knowledge is a human product, evolved socially and individually [4]. The institutional world, however massive and objective it may appear to the individual, is actually a humanly produced, constructed world. This humanly produced world affects the individual mind. The relationship between man, the producer, and the world, his product, is and remains a dialectical one. If this holds also in science, it leads to a realization that man is capable of producing science that he then experiences as something other than human product.

Effective teaching depends a lot on how useful the concepts are to the students, and how concepts are taught shapes the way the information is understood. This is named the knowledge construction process [7]. Knowledge construction process is the procedures by which social, behavioral, and natural scientists create knowledge and how the implicit cultural assumptions, frames of references, perspectives and biases within a discipline influence the way knowledge is constructed within the discipline [8].

In Western countries education and life-long learning have become essential resources for work achievement and

personal development [3]. The critical matter in the Western individualist countries is less to know how to do, than to know *how to learn* [3]. As most information is online, what is really required is the skill to decide what to look for, how to retrieve it, how to process it, and how to use it for the specific task that prompted the search for information [3].

We would like the students to be encouraged to use the systems of knowledge of their own cultures where applicable. This would make their interpretations more accurate, having tangible counterparts in the students' world. Moreover, this way we would not be carrying a message to the students that Western thinking would be superior to the systems of knowledge of their own cultures. Western games, society, and infrastructure, for example, are currently clearly visible in the examples of CS literature.

Deck of cards or chess, family structures, and traffic arrangements, just to mention a few, are used to clarify computational concepts, and these examples may not be culturally meaningful entry points to learning CS for Tanzanian students. We hope to find more pragmatic factors that would support the use of information and communication technology, and more authentic interpretations for computational concepts, especially in non-Western cultures. Education should supply mental support in terms of reducing cognitive overhead and allowing visual imagination through displays [6].

LEARNING TO PROGRAM IN THE TANZANIAN CONTEXT

Context description

The ICT rather increases than decreases the gap between developed countries and less developed countries. Tanzania has been lagging behind in this field for a long time but slowly some development has taken place. Cell phones and the Internet are becoming more and more available for people in the urban areas. People in the cities are discovering the business opportunities that come along by the new ways of communicating. To gain as much as possible from ICT it is important to be not only receivers but also producers [3]. Programming skills empower local people to master ICT, not only to use it, and enable them to fit it into the local or individual context. In Tanzania most people speak Swahili and tribal languages. The knowledge in English is usually very poor among the majority of the population that has not received secondary level education. Even though Swahili is the most spoken language in East Africa there are still very few computer programs and web pages available in Swahili.

The Ministry of communication and transportation in Tanzania has expressed that the Government shall use ICT as a tool for fighting endemic communicable and infectious diseases – especially HIV/AIDS – so as to improve the quality of life of Tanzanians [17]. HIV/AIDS is one of the biggest problems that Tanzania is facing today. At least 1.5 million Tanzanians are estimated to be infected by the virus [9]. The reason that the virus has spread so quickly in many African countries is mainly due to lack of information and education. In countries with generalized epidemics in Africa, up to 80% of women aged 15–24 have been shown to lack useful and meaningful knowledge about HIV/AIDS [11]. Economic Commission for Africa, the Africa Union, and the Southern African Development Community, are tackling HIV/AIDS as a development issue [10].

During spring 2002 a programming course was held at Kidugala Secondary School. Kidugala is a small village in the rural area of the southern highlands of Tanzania. The roads to Kidugala are in poor condition and only two buses a day leaves from/to the village. Neither an Internet connection nor a phone line is available within the compound, which makes the village rather isolated. According to a missionary in Kidugala there are two neighboring spots where cell phones occasionally get reception. Apart from that, radio calls and regular mail are the only way to communicate with the rest of the world.

Fieldwork

Kidugala Secondary School has a *computer laboratory* equipped with eight computers that were provided to the school in 1995. The electricity is taken from the nearby river with a hydroelectric power station. A programming course, including basics in Java programming, was organized for the students. During the course the students were supposed to develop their own material such as programs, web pages and pictures related to their needs. The goal of the current project is to *improve the knowledge of computers and find new ways of how they could be used among students in Tanzania*. To achieve this it is important to figure peoples needs, and what ordinary people could gain from this knowledge. To be able to contextualize the teaching material, it is important that the material is developed and cultivated by the end user group, in this case students in Tanzania.

The idea with this study is to provide the students with the necessary tools to shape a teaching environment adjusted to their particular needs and problems. Students need to identify appropriate tools to program what they need or want. To do this we used a problem based learning approach. Since one of the biggest problems in this area is spreading information about HIV/AIDS, we laid focus on contextualizing this course by *developing digital teaching material concerning this disease.* One problem in the development of contextualized material is that it demands a lot of creativity from the students. Traditional teaching in Tanzania is very teacher-centered and does not encourage creativity. Usually the students are only expected to learn what the teacher tells them without having the chance to explore their own ideas. In working with programming, students have to work creatively, and this is also a reason why we believe programming could play an important role in Tanzanian schools. Since creativity is not encouraged by either society or school, a lot of effort in this study was put in figuring out how to make students work in a creative way.

Since the computers at the school are rather old, an early release of the Java environment had to be installed on all the computers except for one. The programming software used within the course was shareware. On the seven computers running under Windows 3.11 the development kit JDK 1.0.2 (Beta 3) was used. There was also one dedicated computer running Windows 98 for learning a newer version of Java. This was needed for Jeliot, an environment that visualizes the running of Java programs and which has been successfully used especially with novice programmers [12]. The teaching material consisted of the following parts: electronic material produced by two Tanzanian students at University of Joensuu [13], distance education courses targeted at Finnish high school students learning computer science over the web [18], the programming environment JAVA 2 SDK - Standard Edition Documentation Version 1.3.1 by Sun Microsystems; and regular textbooks [14][15][16]. In addition, the teacher developed his own teaching material.

15 out of 91 volunteering students from Form 2, 3 and 4 were selected to participate in the programming course. The regular computer teacher of the Kidugala Secondary also joined the course. All the students had previously taken at least one year of computer studies, however their knowledge of computers was still poor. The lectures lasted for two hours and were held Monday to Friday in the afternoons after the ordinary school lessons. The course started 2nd April and ended 20th May.

The research method used in this study was "action research". In this method the researcher is not just an outsider, but personally involved in the learning process. It includes searching for development and possible change in a concrete everyday environment. The method could be described as a kind of a problem solving process, where evaluation has an important role. Another challenging issue is to interpret and analyze the learning situation in a certain time and place from the gathered information and theoretical background. [20]

The teacher started by using existing teaching material and tried to get as much feedback as possible to improve the teaching and to reply to the students' needs and interests. During the course the teacher received input such as questions, answers, observations and code to analyze. To reach sustainability with the improvements the teacher's diary played an important role as it made it possible to go back and see how changes in teaching were reflected on the results. Observations to be recorded in the teacher's diary in are [21]:

- Summaries of what has been taught.
- Reflections on discussions.
- Questions and answers given by the students.
- Daily and weekly summaries of the process.
- Guesses, hunches, thoughts.
- Reflections on re-reading the diary.

The teacher helped the students to increase their performance through explanations, by giving examples and by handing out problems and questions. By analyzing the students' program code, the visual appearance of their programs, the materials they had created, their questionnaires and the teacher's diary, the main results of the research were acquired. Since the teacher's understanding of concepts differs from that of the students, a cultural encounter occurs. This was also why the action research approach was necessary to improve the teaching by emphasizing the overcoming of cultural barriers.

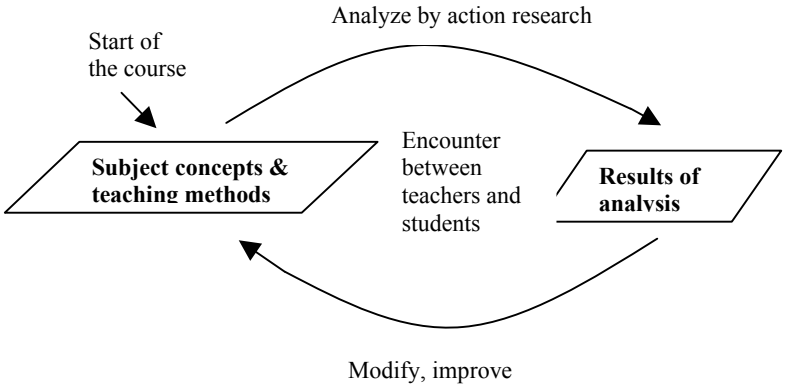

Figure 2: Contextualization cycle of teaching programming

Course outcomes

As mentioned before, much of the focus of this study was to find out how to construct a creative environment for the students to work in. This contextualization process of teaching programming is sketched in Figure 2.

At the beginning the teacher used *traditional teaching methods* by using the blackboard and taking simple examples from various textbooks.

As it was noticed that the students had problems in understanding concepts of programming, the visualization tool Jeliot was used. This improved the *understanding of the concepts* of programming but did not increase the students' creativity.

To make the students work more creatively in developing their own programs, the students were taught how to write simple applets by using the graphical tools that the Java environment provides. While working with applets a major increase in the students' creativity was noticed, since *programming now began to make sense for them.* The students felt that they actually could create something else than just a number changing value or other quite meaningless exercises.

But what really made a significant difference in the students' creativity was when they started to develop material for a web page containing information about HIV/AIDS. This was something that all the students could

relate to and see the importance of. By developing this web page *both the programming skills and the students' creativity increased*. This was noticed from all the ideas of programs that the students came up with and how they tried and managed to solve these ideas. While students worked with the material for the web page, they also improved their knowledge about the disease.

HIV/AIDS education is closely related to education on sexual matters, a topic that is a taboo in Tanzania and usually difficult to talk about. However, during the programming course, it was noticed that computers could work as a neutral platform in dealing with questions concerning the disease.

Before the course only a few students knew the potential of Internet and none of them had used it. Still they managed to develop a webpage containing pictures, graphics, applets, and text, offering information related to the disease and the problems that come with the disease in this area. The students were free to use either Swahili or English in their material although they where encouraged to use Swahili or to produce the information in both languages. Considering the students' poor background in using computers and especially in programming, the students' results were good [22]. At the end of the course the students knew the basics of programming in Java, had increased their knowledge about HIV/AIDS and had had an opportunity to explore their creative side.

Figure 3: An Applet made by a student for the web page about HIV/AIDS [19].

Figure 3 shows an animation with the message, "if you don't know enough about the disease you might end up dead".

Discussion

Course schedule. To reach even better results during the course the students should have had more time to develop their own material. Since the course was held in the afternoons after ordinary classes, the students were often tired and not as active as they could have been otherwise. The students also had to sacrifice other activities such as sports, singing in the choir or doing homework. These activities were usually held at the same time. The schedule was tight as the teacher wanted the students to learn and produce as much as possible during the short course. The teacher did not spend a lot of time reviewing previous lessons, which was difficult for students who missed some lessons. Togheter with the fact that some students were not interested enough this led to several dropouts. At the end of the course only half of the initial number of students still attended the lessons. One solution would have been to give the course during school breaks or in weekends so that the students could have focused on the course only.

Programming environment. Another problem that gave the students less time to work on their programs was that the computers were not powerful enough for the programming environment and it took a long time to compile the programming code. To solve that problem either newer computers would be needed, or it would be necessary to switch to another programming environment, including a different programming language. During the course the teacher occasionally handed out simple problems for the students to verify that they had understood certain concepts. The results were unsatisfactory. However, using the Internet seemed to increase their motivation as they understood that their material would be available globally. This was also the reason to teach programming in Java.

Problem based learning. To increase the creativity among the students they could have been given more problems or thinking exercises to work with. Often the students took the teacher's examples and tried to modify them instead of using their knowledge to create own examples. Because the students in Tanzania are used to just memorizing what the teacher tells them it is necessary to apply problem based approach to intensify the learning outcomes.

Sustainability. The work with a webpage about HIV/AIDS was very successful for the students' results. But there is still more work to be done, so that it could be used as an information page for others. However, the teacher made it available on the Internet [19]. Students planned to continue the work after the teacher left Kidugala, but unfortunately that has not happened yet.

PEDAGOGICAL OPPORTUNITIES OF CONTEXTUALIZED PROGRAMMING STUDIES

Our first teaching period in Tanzania was already in 1995, on implementing multimedia version of a Tanzanian folklore story. This story was animated using Visual Basic. The results were impressive. Period in 2002 was on Java education. There is a long step from Visual Basic to Java. But according to the last experiences even Java is a useful tool as it opens the doors of the Internet world, which helps to motivate the students.

The focus has been to identify meaningful entry points to learn programming rather than just only to choose an appropriate programming environment from a selection of western (technical) markets.

This kind of context – a meaningful programming link – develops knowledge and learning skills. When one is

programming a Java application in the digital environment, he has simultaneously a practical goal in his real life. The goal could be shared socially and some parts of the construction process of the solution, too. The goal forces to design a solution. One can probably copy something from others' ideas, but the real, final solution is in the student's own hands. This self-direction is essential and develops the problem solving skills all the time.

The development of the problem solving skills has an impact to other parts of life as well. It develops responsibility and fosters creativity. When this is linked to the mastery of high technology skills, the learning process serves not only an individual student, but has also impact to her cultural and social context on the whole.

In the future we will continue our contextualization efforts in a wider scale. A motivating topic like AIDS is not enough – we will concentrate on the conceptual level and new teaching methods. We will consider specific ICT curriculum needs in Tanzania. We cannot just translate a western curriculum for instance to Tanzania. Some courses should be replaced by new ones. The whole curriculum has to be designed from the Tanzanian point of view. This does not imply a lower quality, but probably quite a different perspective. This perspective might give the possibility to a bi-directional learning process – novel opportunities for Tanzanian students and re-contextualized teaching methods for their Western counterparts.

CONCLUDING REMARKS

Our findings, analyzed by action research methods, indicate that teaching of programming gains substantially from culturally relevant entry points. In our case, the culturally meaningful task to design digital material and small-scale games for HIV/AIDS education increased the motivation of the students to learn the required technical tools.

Two fundamental goals and driving forces of education are creativity and self-direction. Although various visualization tools improved the students' comprehension of programming concepts, their inspiration and imagination were sparked only by a contextual exercise. This suggests that choosing an appropriate language for novice programmers is less important than providing them with a meaningful application.

Applying the idea of ethnocomputing at a more conceptual level remains a question of further research. In this study, we applied it to choosing a motivating problem to foster the learning of a particular programming language, Java. In the future, we will be looking into how various computational concepts are understood and communicated in a given cultural environment. This will hopefully open us conceptual entry points to teaching programming, in addition to the motivating entry points highlighted by the present study.

REFERENCES

[1] D'Ambrosio, U, "Introduction: Ethnomathematics and its First International Congress," *ZDM: Zentralblatt für Didaktik der Mathematik* 1999 31, 2#, pp. 50-53.

[2] Hofstede, G., *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and its Importance for Survival*. McGraw-Hill, 1997.

[3] Castells, M., *The Internet Galaxy. Reflections on the Internet, Business, and Society*. New York: Oxford University Press, 2001.

[4] Berger, P.L., and Luckmann, T., *The Social Construction of Reality*. Allen Lane, 1966.

[5] Kuhn, T. S., *The Structure of Scientific Revolutions*. Chicago IL: The University of Chicago Press, 1962.

[6] Tedre, M., Kommers, P., and Sutinen, E., "Ethnocomputing: Considering ethnological factors in Computer Sciences," in *Exploring Cultural Perspectives: Integration and Globalization*, eds. Wyness, M. and Richardson, A., Edmonton: ICRN Press, 2002.

[7] Banks, J. A., *An Introduction to Multicultural Education, 2nd Edition.* Allyn and Bacon, 1999.

[8] Banks, J. A., "Multicultural Education: Historical Development, Dimensions and Practice," in *Review of Research in Education* 19: pp. 3-49, 1993

[9] UNAIDS/WHO, Epidemiological Fact Sheets on HIV/AIDS and sexually Transmitted Infections, 2002, Available at http://www.who.int/emc-hiv/fact_sheets/pdfs/Tanzania_EN.pdf

[10] UNAIDS, Fact sheets 2002 Sub-Saharan Africa, Available at http://www.unaids.org/worldaidsday/2002/press/factsheets/FSAfrica_en.doc

[11] UNAIDS/WHO, AIDS Epidemic Update December 2002, Available at http://www.unaids.org/worldaidsday/2002/press/update/epiupdate2002_en.doc

[12] Ben-Ari, M, Myller, N, Sutinen, E, Tarhio, J, Perspectives on program animation with Jeliot. Software Visualization. Dagstuhl Castle, Germany, Lecture Notes in Computer Science 2269, 2002, 31-45.

[13] Laisser, E. & Lupilya, E. *Introduction to JAVA*. URL: http://cs.joensuu.fi/~mvirnes/Programming1/index.htm (the site visited 10th Feb. 2003)

[14] Wright, C. (2001) *Teach Yourself JAVA,* Second Edition. Hodder & Stroughton. McGraw Hill – NTC.

[15] Budd, T. (2000). *Understanding Object-Oriented Programming with JAVA*. Oregon State University. Addison Wesley.

[16] Chapman, S., J. (1999). *Introduction to JAVA*. Prentice Hall

[17] Ministry of communication and transportation, National ICT Policy of Tanzania May 2002, available: http://www.ethinktanktz.org/esecretariat/DocArchive/zerothorder.pdf

[18] Haataja, A., Suhonen, J., Sutinen, E. & Torvinen, S. (2001). High School Students Learning Computer Science over the Web, Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (IMEJ) 3(2), October 2001.

[19] Ukimwi Unuaua Jihadhari, Aids Project Kidugala Seminary in Spring 2002. URL: http://home.bip.net/duveskog/AIDS/ (the site visited 28th March 2003)

[20] Winter, R. (2001) Action Research, Relativism and Critical Realism: A 'Theoretical' Justification for Action Research. Collaborative Action Research Network. http://www.did.stu.mmu.ac.uk/carn/Members_papers/Richard_Winter.html (the site visited 28th March 2003)

[21] Partially from Hughes, I, "How to Keep a Research Diary", Action Research Electronic Reader, The University of Sydney, 1996, Available at http://www.scu.edu.au/schools/gcm/ar/arr/arow/rdiary.html (the site visited 28th March 2003)

[22] Chaltu, G., Duveskog, M., Sutinen, E. & Vesisenaho, M. "HIV/AIDS education in Tanzania blended with a programming course". Submitted, 2003.

[23] Guba, E.G., Lincoln, Y.S., "Competing Paradigms in Qualitative Research", *Handbook of Qualitative Research* (eds. Denzin, N.K., Lincoln, Y.S.) SAGE Publications, 1994:105-117.