Modeling 3D Interactive Environments for Learners with Visual Disabilities

J. Sánchez and N. Baloian

Department of Computer Science, University of Chile
Blanco Encalada 2120, Santiago, Chile
{jsanchez, nbaloian}@dcc.uchile.cl

Abstract. Educational software has been criticized for not using explicit models to generalize and replicate good practices. Actually almost every educational program has a model, but most of them remain implicit. In this paper we propose a methodology for developing educational software for children with visual disabilities. Multimedia software for these children has some particularities reflected on our model with emphasis on process modelling including learner evaluation and feedback. The model emerges from research on developing educational software for children with visual disabilities and studies concerning the design of educational software for sighted learners. The model was validated by special education teachers and software designers trying the model with five software products based on model heuristics.

1 Introduction

The design of educational software should make extensive use of all multimedia capacities modern computers can offer. This implies that users with visual disabilities are unable to access to this software because of the use of graphical user interfaces.

Although not as much as for sighted people, there have been some developments of educational software for people with visual disabilities. However, they usually lack of critical interface elements commonly present in software for sighted children. Most software does not include explicit model knowledge and skills learners should construct when using the software, an explicit learner model, and the implementation of appropriate feedback to improve the learners' performance. To many authors, designers of educational software for children with disabilities conceive the software with interaction restrictions in their minds, fixing the interaction modes from the very beginning. Thus software is from the beginning conceived with limitations. We propose that educational software for learners with visual disabilities should be designed without taking into account from the beginning the users' disabilities. They should start by considering relevant modeling aspects. Only when it comes to the point of mapping the inputs and outputs of models into an interface, the learner capabilities and disabilities should be taken into consideration to map these variables on proper devices.

Since educational software development process depends on people, tools, and methodologies involved, and considering that we have not a clear methodology to carry out this process for children with visual disabilities, the results mainly depends on the skills of the involved people. This can cause many drawbacks typical for a hand-crafted process. Software engineering uses methodologies to help to reduce the craftsmanship level of software development by using the best methodological practices.

There have been some proposals for methodologies to develop educational software (Alessi & Trollip, 2001; Soares, 2001; Dillenbourg & Self, 1992) and courseware (Baloian et al., 2001). An interesting question is whether we need another new methodology for this kind of software or if the existing one can work appropriately. We think the process should start from existing, well accepted methodologies extending and adapting them to this particular case in order to guide the implementation team to think about the possibility of including elements of intelligent tutoring systems in software aimed to blind users. The goal of this methodology is to assist developers in considering critical components for educational software design. The methodology proposes also a characteristically system architecture.

Software based on virtual environments for users with visual disabilities are based on the presentation of graphic information by text-to-speech translation that reads Web pages displayed through browser and three-dimensional spaces of navigation environments with sounds that can get close, far or move to mentally represent the space (Mereu & Kazman, 1996) and to develop cognitive skills (Savidis et al., 1996). This can be seen in Morley et al. (1998), where blind people develop a special way of navigating through a known environment and represent spatial structure with cognitive difficultness. The system uses different output devices such as concept keyboard, tablets, switches, tactile interfaces (Lange, 1999), and forcefeedback (Ressler & Antonishek, 2001).

The HOMER UIMS was produced by Savidis and Stephanidis (1995), Savidis et al., (1996) developing dual interfaces to integrate blind and sighted learners. HOMER integrates visual and non visual interaction with objects and their relationships. The browser BrookesTalk reproduces a Web page by using synthesized voice with words, sentences, paragraphs, and offering different points of view of the page to simulate scanning (Zajicek et al., 1998).

A game for audio concentration is presented by Roth et al., (2000) consisting of pairing different levels of geometric figures, basic, and derives. To represent geometric figures graphically they constructed a bi-dimensional sound space. This concept allows graphic representation such as icons to be represented by the perception of moving sounds in the spatial plane. Blattner and Brewster introduced "earcons" as non verbal audio messages to provide information to users about computer objects, operation, and interactions (Blattner et al., 1998; Brewster, 1998). Each dimension corresponds to a musical instrument and the points of the plot correspond to pairs of frequency in a scale. The horizontal movements from left to right are equivalent to a frequency variation of the first instrument and the vertical movement to frequency variations of the second one.

AudioDoom (Lumbreras & Sánchez, 1999) allows blind children to explore and interact with virtual worlds by using spatial sound. The game was based on the traditional Doom game where the player moves through corridors discovering the environment and solving problems simulated with objects and entities that inhabit a virtual world. VirtualAurea (Sánchez, 2001, 2002) was developed after it was proved that sound-based virtual environments can help to develop tempo-spatial cognitive structures of blind children. It is a spatial sound tool editor that can be used by parents and teachers to design a wide variety of spatial maps. Users can integrate different sounds by associating them to objects and entities in a story. AudioMemory (Sánchez & Flores, 2004) a virtual environment based on audio to develop and use short-term memory. It was also modeled with mathematics contents, AudioMath, to assist the learning of basic mathematics operations such as multiplication and division. Results evidenced that both software helped to develop and enhance memory and mathematics learning in blind children. AudioChile (Sánchez & Sáenz, 2005) is a 3D interactive environment for children with visual disabilities to help them to solve problems related with the Chilean geography and culture. AudioChile can be navigated through 3D sound to enhance spatiality and immersion throughout the environment. 3D sound is used to orientate, avoid obstacles, and identify the position of diverse personages and objects within the environment.

2 Process Modeling

The methodology proposed is based on the following hypotheses: 1. The knowledge and skills a learner has to develop with the aid of the software are measurable and can be represented. This implies that a learner's model can be constructed, and 2. The software represents an interactive environment (real or imagined) for learning. This means that the software allows the construction of knowledge by the learner

The "workflow" for developing educational software for learners with visual disabilities proposed by this methodology is depicted in the Figure 1. To explain this methodology we use AudioDoom (already introduced in chapter 1). Normally, a software development process starts with the definition of software requirements. In this case, the requirements are represented by the learning goal that in the case of AudioDoom is the ability to create a mental model of the surrounding environment. According to the learning goal, an appropriate scenario should be conceived to allow learners to develop a skill or knowledge. In AudioDoom the idea was having the learner to discover a labyrinth full of sound emitting objects and entities. The next step is modeling the environment. At the same time, and based on artificial intelligence strategies, the learner's knowledge should be modeled (Baloian et al., 2002). Developing and describing a model has its own process (Zeigler, 1976). The result of this step is a formal model description for both, the learning environment and the learner's knowledge in paper. A computer program has to implement this. At this point, it is important to consider the development of an (or use an already existing) editor for generating different environments, such as editors for constructing different labyrinths, instead of having a single environment "hardwired" represented by the program. After this process, model input and output variables are clearly identified. Then we need to map or project them on input/output devices suitable for children with visual disabilities.

As we see in Figure 1, cognitive goals will not only influence the definition of learning environments but also the generation of metrics for evaluating knowledge construction by the learner. This will be discussed in more detail in the following chapter. Then learners should explore the environment as a way of testing. Test results should be evaluated through usability methods and determine the effectiveness and impact to help learners to achieve the cognitive goals. The evaluation may cause a revision of the real world representation, the model, and the interface. Revisions of the real world and its modeling can be mostly caused by the failure of the software's effectiveness in supporting learners to achieve the cognitive goals: the environment does not provide the adequate learning activities and the model does not implement them properly. Revisions of the interface (projection of the input/output values on adequate devices) can be caused by usability drawbacks in the software.

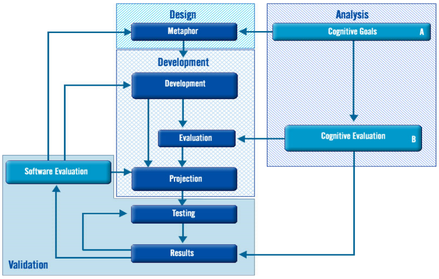


Fig. 1. The workflow for developing educational software for learners with disabilities

Analysis: Consists of two sub-stages, A and B. The first stage is to define cognitive goals to be achieved by the learner. This corresponds to the definition of software requirements. The second stage is to define procedures and functions to evaluate the achievement of cognitive goals. *Design***:** In this stage a metaphor will be defined for a "world" or scenario where the learner constructs knowledge through the interaction with this world. Normally, this is game type of software and "playing rules" are defined. This leads to define the model of the world and the knowledge to be constructed. *Development:* Consists of three sub-processes. The first process is the computational implementation of models of the world and learner. We recommend exploring the possibilities of developing editors for implementing different scenarios of the same "world". The second stage is the implementation of the evaluation process and the feedback to the student. The third stage is the projection of the models. We identify input and output variables of models as well as parameters and results (including the feedback to the learner) of the evaluation function. These values have to be "projected" properly over the haptic, audio, and visual (for people with residual vision) input/output devices available. We verify a wide variety of input/output devices to avoid limiting to traditional devices such as joysticks and keywords for input and sound for output. Haptic devices such as tablets, electronic boards, and Phantom can give blind users sensations of being "touching" virtual objects. It is important to make these actions after setting the models in order to ovoid restricting from the beginning of software design. Some guidelines to implement the projection are given below based on the literature and our own experience in developing software for blind children. *Validation:* Consists of two sub-processes. First, we develop usability tests to get data about how well the system fit our objectives in order to attain the cognitive goals set at the beginning. We emphasize the analysis of some elements of human-computer interaction. Second, we analyze these results and study how the metaphor, models, and the projection of input/output variables can be improved. Normally an error in the integrity of the system for learning can imply to review the metaphor and models used. Usability issues can lead to review the projection.

3 Modeling the Resulting Architecture

Fig.2. represents the architecture of the resulting software containing the following components:

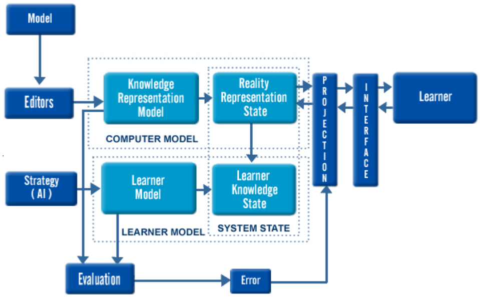


Fig. 2. Architecture of the resulting software

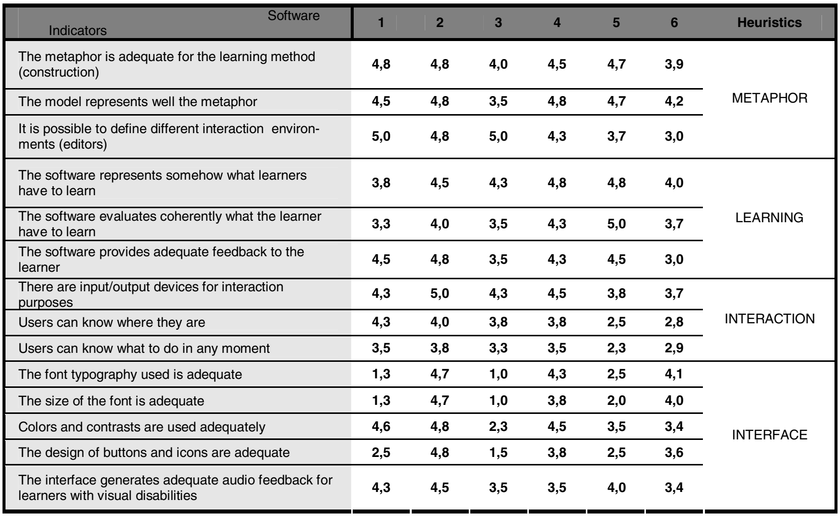
Metaphor of the real world (Model): According to the cognitive skills to be developed real world metaphors are designed as well as the activities learners have to do to attain the cognitive goals by considering their interest and motivation. *Editor:* Tools to construct an internal model based on 2D/3D graphic representations or auditory representations. *Computer representation of the real system:* Corresponds to the computational representation of the problem, the real world metaphor, and models the knowledge. Contains functions, parameters, and variables of the state of the system describing the situation of the represented world and how the transition from one state to other will be made by considering the interaction of the learner with the software and reflected on the entry variables. *Strategies*: This component gives the strategies to be used to model the state of the learner's knowledge. They are taken from the field of artificial intelligence applied to intelligent tutoring. Some of them are the overlay model (Kass, 1989) that treats the learner's knowledge as a subset of an expert knowledge. The Differential model (Clancey, 1987) extends the previous model by dividing the learner's knowledge in two categories: knowledge that the learner should know and knowledge that is not expected to be known by learners. The perturbation model (Kass, 1989) supposes learner should posses a potentially different knowledge in quantity with respect to an expert. The model can represent the knowledge and beliefs of the learner beyond the ranks of the expert's model. *Learner model*: This component represents what the system thinks about the state of the students learning in a certain point. It contains knowledge and skill representations the learner should construct, the variables of the state of learner representing the level of learning in a certain moment, and the rules about how to upgrade this information given the interaction with the system and reflected in the change from one real world model state to another. Thus the learner model is given by making inference of the individual knowledge by analyzing the performance (Dillenbourg & Self, 1992). *Evaluation:* This component defines the difference between the knowledge model represented in the software and the knowledge model of the learner generated by the strategies. Thus an error measure is produced and projected to the interface as student's feedback. *System* *projection*: This is the main component to certify that the software can be fully assimilated by children with visual disabilities. It is in charge of projecting most interactions, state variables, and feedbacks from and to the software.

4 Model Evaluation

We tested our model with three special education teachers and two software developers. They evaluated the model by analyzing five products to check how well they meet the methodology proposed above. To do this we designed a Likert type scale based on model heuristics. From the model proposed above we defined four major heuristics for evaluation purposes: metaphor, learning, interaction, and interface. Metaphor included adequacy to the mode of learning, how well it represents the model, and if it defines different interaction environments (editors). Learning includes if the software represents what learners have to learn, evaluates learning adequately, and provides feedback to the learner. Interaction includes if the input/output devices are adequate, and how well users can orient and know what to do and where to go by themselves. Interface included font and typography, colors, buttons, icons, audio cues, and feedback used.

The results of the model evaluation are presented in Table 1. Possible answers spanned from "do not meet the heuristic" (1) to "highly meet the heuristic" (5). Average resulting scores were from 3.4 (VirtualAurea) to 4.6 (AudioMemorice), evidencing that most software analyzed meet the minimum standards posed by the model.

Table 1. Model evaluation results. 1. AudioBattleShip, 2. AudioMemorice, 3. VirtualAurea, 4. Theo&Set, 5. CantaLetras, 6. AudioVida.



Software Indicators	1	2	3	4	5	6	Heuristics
The metaphor is adequate for the learning method (construction)	4,8	4,8	4,0	4,5	4,7	3,9	METAPHOR
The model represents well the metaphor	4,5	4,8	3,5	4,8	4,7	4,2	
It is possible to define different interaction environments (editors)	5,0	4,8	5,0	4,3	3,7	3,0	
The software represents somehow what learners have to learn	3,8	4,5	4,3	4,8	4,8	4,0	LEARNING
The software evaluates coherently what the learner have to learn	3,3	4,0	3,5	4,3	5,0	3,7	
The software provides adequate feedback to the learner	4,5	4,8	3,5	4,3	4,5	3,0	
There are input/output devices for interaction purposes	4,3	5,0	4,3	4,5	3,8	3,7	INTERACTION
Users can know where they are	4,3	4,0	3,8	3,8	2,5	2,8	
Users can know what to do in any moment	3,5	3,8	3,3	3,5	2,3	2,9	
The font typography used is adequate	1,3	4,7	1,0	4,3	2,5	4,1	INTERFACE
The size of the font is adequate	1,3	4,7	1,0	3,8	2,0	4,0	
Colors and contrasts are used adequately	4,6	4,8	2,3	4,5	3,5	3,4	
The design of buttons and icons are adequate	2,5	4,8	1,5	3,8	2,5	3,6	
The interface generates adequate audio feedback for learners with visual disabilities	4,3	4,5	3,5	3,5	4,0	3,4	

From the results displayed we can state four initial conclusions. First, we validated the model by evidencing that using heuristics is a clear methodology for model analysis in educational software. Second, all products considered the heuristics in different degrees. Third, metaphor and learning are the heuristics that best meet the standards of our model. Fourth, interaction and interface were the least attained heuristics. Then our model was initially validated with existing educational software through walk-through techniques used by teachers and software developers.

5 Discussion and Further Work

We present a methodology for developing educational software for children with visual disabilities. The model is the result of a growing need for models to develop, replicate, evaluate, and improve educational software for this population. This is a process model with the resulting architecture including ways of evaluating and giving feedback to learners, as well as to set qualitative differences for children with and without visual disabilities. We describe formally and operationally the model and propose some guidelines to design educational software for children with visual disabilities by discussing main generic attributes to include in this software.

The model was tested for viability in educational software design. Interesting results came out when teachers and software developers went through existing educational software for blind children by using some heuristics drawn from the model. Most software did meet the heuristics in different degree. Interactivity and interfaces were the least ranked, meaning that these heuristics need to be carefully considered when design software for children with visual disabilities. Now we need first to improve our heuristics and evaluation instruments, and then apply them to different learning contexts for children with visual disabilities. The next step will be to design and develop software for children with visual disabilities by following step by step the methodology proposed here. Finally, we expect to contribute to the field with an explicit and functional model that can be generalized and replicated to help to improve the learning of children with visual disabilities.

Acknowledgements

This report was funded by the Chilean National Fund of Science and Technology, Fondecyt, Project 1030158.

References

1. Alessi, S. & Trollip, S. (2001). Multimedia for Learning, Methods and Development. Boston: Allyn and Bacon
2. Baloian, N., Ochoa, S. & Fuller, D. (2001). A model for component-based courseware development. Proceedings of the KSI Conference SEKE'01. Buenos Aires, Argentina, Jun. 13-15. KSI Press. USA pp. 363-370
3. Baloian, N., Luther, W., Sánchez, J. (2002). Modeling Educational Software for People with Disabilities: Theory and Practice. Proceedings of the ACM ASSETS 2002 Conference, pp. 111-117
4. Blattner, M., Sumikawa, D. & Greenberg, R. (1998). Earcons and icons: Their structure and common design principles. Human-Computer Interaction, 4, (1) (1998), pp. 11-44
5. Brewster, S.A (1998). Using earcons to improve the usability of a graphics package. Proceedings of HCI'98, Sept. 1-4(1998) Sheffield Hallam University, Sheffield, UK
6. Clancey, W.J.(1987). Methodology for Building an Intelligent Tutoring System. En Kearsley, G., Artificial Intelligence and Instruction - Applications and Methods, pp.193-227

7. Dillenbourg, P. & Self, J. (1992). A Framework for Learner Modeling. Interactive Learning Environments, 2(2), pp. 111-137, 1992
8. Kass, R.(1989). Student Modeling in Intelligent Tutoring Systems –Implications for User Modeling. In A. Kobsa & W. Wahlster, Eds. User Models in Dialog Systems, pp. 386-410.
9. Lange, Max O. (1999). Tactile Graphics – as easy as that. CSUN's 1999 Conference Proceedings. March 15-20, Los Angeles, USA
10. Lumbreras, M., & Sánchez, J. (1999). Interactive 3D Sound Hyperstories for Blind Children. Proceedings of the ACM CHI '99, Pittsburg PA, USA (1999), pp. 318-325
11. Mereu, S.& Kazman, R. (1996). Audio enhanced interfaces for visually impaired users. Proceedings of the ACM CHI 96, pp. 72-78
12. Morley, S., Petrie, H., O'Neill A.-M. & McNally, P. (1998). The use of non-speech sounds in a hypermedia interface for blind users, in Edwards, A.D.N., Arato, A., and Zagler, W.L.(Eds.)'Computers and Assistive Technology'. Proceedings of ICCHP'98. XV, IFIP World Computer Congress, pp. 205-214
13. Ressler, S. & Antonishek, B. (2001). Integrating Active Tangible Devices with a Synthetic Environment for Collaborative Engineering. Proceedings of the 2001 Web3D Symposium. Paderborn, Germany, February 19-22, pp. 93-100
14. Roth, P., Petrucci, L., Assimacopoulos, A. & Pun, Th. (2000). Concentration Game, an Audio Adaptation for the blind. CSUN 2000 Conference Proceedings, Los Angeles, USA.
15. Sánchez, J. (2001). Interactive virtual acoustic environments for blind children. Proceedings of ACM CHI ´2001, Seattle, Washington, April 2-5, 2001, pp. 23-25
16. Sánchez, J. (2002). VirtualAurea: Perception through spatialized sound. Proceedings of ICDVRAT, pp. 67-73. September 18-20, 2002, Veszprém, Hungary
17. Sánchez, J., Baloian, N. & Hassler, T. (2004). Blind to Sighted Children Interaction through Collaborative Environments. Lecture Notes in Computer Science 3198, Springer-Verlag, pp. 192-205
18. Sánchez, J.,Flores, H. (2004). Memory enhancement through audio. ACM SIGACCESS Accessibility and Computing Archive Issue 77-78, pp. 24-31, Sept. 2003-Jan. 2004
19. Sánchez, J., Sáenz, M. (2005). 3D Sound Interactive Environments for Problem Solving. Proceedings of The Seventh International ACM SIGACCESS Conference on Computers and Accessibility, Assets 2005, Baltimore, Maryland, USA, October 9-12, pp. 173-178
20. Sánchez, J., Baloian, N. (2005). Modeling Audio-Based Virtual Environments for Children with Visual Disabilities. Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications ED-MEDIA 2005. Montreal, Canadá. June 27 July 2, 2005, pp.1652-1659
21. Savidis, A. & Sthphanidis, C. (1995). Developing dual user interfaces for integrating blind and sighted users: HOMER UIMS, ACM CHI'95 Conference Proceedings, Denver, CO (1995), pp. 106-113
22. Savidis, A., Sthphanidis, C., Korte, A., Crispie, K. & Fellbaum, K.(1996). A generic direct-manipulation 3D-auditory environment for hierarchical navigation in non-visual interaction. Proceedings of the ACM ASSETS 96, pp. 117-123
23. Soares, M.P.S. (2001). Modeling educational software for teaching quality control with emphasis on quality evaluation software, MAQS. Proceedings of the International Symposium on Computing in Education, Aguas Calientes, México
24. Zajicek M., Powell C. & Reeves C. (1998). A web navigation tool for the blind. Proceedings of the ACM ASSETS'98, 3rd ACM/SIFCAPH Conference. on Assistive Technologies, Los Angeles, USA (1998), pp. 204-206
25. Zeigler, B. P. (1976). Theory of Modeling and Simulation, Wiley, New York, USA, 1976