Université Pierre et Marie Curie - Paris 6

Interaction et usages des modalités non visuelles, accessibilité des contenus complexes

Sélection de travaux personnels

présenté et soutenu publiquement le 7 décembre 2010

pour l'obtention de l'

Habilitation à Diriger les Recherches
de l'Université Pierre et Marie Curie - Paris 6
(spécialité informatique)

par

Dominique Archambault

Composition du jury

Président :	Annick Valibouze	Professeure à l'Université Pierre et Marie Curie - Paris 6
Rapporteurs :	Viviane Durand-Guerrier	Professeure à l'Université Montpellier 2
	Jan Engelen	Professeur à l'Université Catholique de Louvain
	Jaime López-Krahe	Professeur à l'Université Paris 8
Examinateurs :	Pascal Estraillier	Professeur à l'Université de La Rochelle
	Klaus Miesenberger	Professeur à l'Université Johannes Kepler de Linz
	Stéphane Natkin	Professeur du CNAM
	Christian Retoré	Professeur à l'Université Bordeaux 1
Invité :	Damien Olivier	Professeur à l'Université du Havre

Équipe « *Interfaces Non Visuelles et Accessibilité* » — Faculté d'ingénierie (UFR 919)

Mis en page avec la classe thloria, modifiée par l'auteur.

Résumé

Le besoin d'accessibilité est de plus en plus prégnant dans nos sociétés, et l'accessibilité réelle progresse de façon indéniable dans différents domaines, quoique très lentement. Dans la société numérique, les "nouvelles" Technologies de l'Information et de la Communication portent en elles l'espoir de progrès formidables en matière d'accessibilité. D'ailleurs Tim Berners-Lee, inventeur du Web, voulait dès sa création *"mettre le Web et ses services à la disposition de tous les individus, quel que soit leur matériel ou logiciel, leur infrastructure réseau, leur langue maternelle, leur culture, leur localisation géographique, ou leurs aptitudes physiques ou mentales".* Cet objectif s'étend à nos yeux bien au delà du web et englobe toute l'information numérique. Malheureusement la grande vitesse de développement des TIC laisse peu de place à la réflexion sur leur accessibilité, et les acteurs impliqués se focalisent sur les interfaces-utilisateur dominantes, augmentant ainsi la fracture numérique.

Notre activité se situe dans l'étude des modèles informatiques permettant de tirer parti de ces technologies pour améliorer l'accès de tous à la société numérique, et dans l'intégration des principes de ces modèles spécifiques dans les modèles utilisés pour les applications grand public, ce qui peut être une façon de définir le terme "accessibilité numérique".

Si les documents électroniques sont aujourd'hui prévus pour être utilisés dans le cadre d'interfaces multimodales, il s'agit bien souvent d'interfaces spécifiques à l'application et non à l'utilisateur et auxquelles ce dernier doit s'adapter (la plus courante étant bien sûr écran/clavier/souris). L'utilisation de modalités alternatives non visuelles comme le braille, la synthèse vocale ou des interfaces tactiles nécessite une conversion de modalités qui pose bien souvent des problèmes, dus en particulier au fait que les modèles de représentation de données sont trop souvent liés aux modalités principales utilisées.

Pour pouvoir exploiter pleinement le potentiel offert par les interfaces non visuelles, nous travaillons sur des modèles de représentation indépendant des modalités, qui permettent d'adapter l'interface aux besoins de l'utilisateur, en particulier à ses capacités visuelles. Les déficiences visuelles sont multiples et parfois liées à des troubles divers (cognitifs, communication, moteurs, auditifs). Les modalités non visuelles ont néanmoins deux points communs très forts : leur aspect éminemment linéaire, et leur incapacité à offrir une perception globale. Notre objectif est de développer des outils qui s'adaptent automatiquement à l'utilisateur, en fonction de ses spécificités. D'abord bien sûr en fonction des modalités qu'il utilise, mais aussi en fonction de ses compétences, notamment dans le cas des enfants. Cette adaptabilité est rendue nécessaire dans le cas d'une utilisation pédagogique : adaptation à la progression, aussi bien dans les disciplines scolaires que dans l'utilisation des outils. Elle l'est plus encore dans le cas d'une utilisation éducative avec de très jeunes enfants ou des enfants ayant des difficultés supplémentaires (cognitives, de communication, d'éveil), pour lesquels il est important de les faire entrer dans une dynamique de progrès.

Les travaux présentés dans ce mémoire concernent donc l'accessibilité numérique, et en particulier l'accessibilité à des contenus que nous appelons complexes car ils sont structurés et/ou composites, et constitués d'éléments de différentes natures. Les domaines d'application que nous avons étudiés sont les sites Web, les expressions mathématiques et les jeux vidéo. Après une introduction et une présentation du contexte, dans laquelle nous nous attachons à décrire un certain nombre de problèmes concrets touchant à l'accès à des contenus complexes, ainsi que l'état de l'art de ce domaine, nous présentons dans le mémoire une sélection de travaux organisés autour de 4 parties complémentaires, chacune d'entre elles correspondant à un ou plusieurs de nos domaines d'applications. La première partie concerne les interfaces utilisateurs adaptées, permettant de tirer au mieux partie des modalités spécifiques utilisées par les personnes handi-

capées visuelles. La seconde partie s'attache aux outils collaboratifs transmodaux, c'est-à-dire permettant la collaboration autour du même contenu entre personnes utilisant des modalités différentes, afin d'échanger, de travailler ou de jouer ensemble. La troisième partie décrit des outils de transcription que nous avons développé permettant de convertir des contenus formatés pour une modalité donnée dans des modalités alternatives. Enfin dans la quatrième partie, nous présentons une réflexion sur des modèles de représentation accessibles, c'est-à-dire des modèles contenant l'information nécessaire à la présentation des contenus selon différentes modalités, capables donc de faire fonctionner les interfaces spécifiques et/ou collaboratives présentées dans les deux premières parties, en utilisant si nécessaire les outils de transcription présentés dans la troisième partie. Cette réflexion doit être menée conjointement aux travaux actuels sur les standards de représentation de scenarii multimédia et de documents scolaires. Dans le domaine des jeux vidéo nous tentons d'étendre les résultats obtenus dans le projet TiM sur l'étude des interactions de type jeux vidéo et leur adaptation aux modalités non visuelles, afin de proposer un modèle de framework d'accessibilité permettant de créer des jeux accessibles à tous ; c'est-à-dire des jeux que les utilisateurs de modalités alternatives pourront utiliser, fût-ce au travers d'interfaces de jeu spécifiques permettant d'accéder au jeu original et de le contrôler de façon efficace.

Le document se termine par un court chapitre intitulé "Accessibilité et Société" dans lequel nous discutons de l'accessibilité dans la société française d'aujourd'hui, et des forces et limites des outils techniques pour faire avancer l'idée de l'accessibilité. L'accessibilité, dans tous les domaines, permet non seulement d'améliorer la vie des personnes en situation de handicap, en leur permettant d'accomplir de façon autonome des actes qui leur demandait auparavant une assistance permanente, mais aussi d'améliorer la vie de chacun, par une meilleure adaptation des outils aux personnes et aux ressources disponibles. Il est bien évident que l'existence d'outils accessibles n'est pas suffisant pour rendre la société plus accessible, mais c'est une condition nécessaire et nous croyons qu'ils y contribuent.

Abstract

The need for accessibility is growing in our societies, while real accessibility is truly progressing in several domains, but very slowly. In the digital society, the "new" Information and Communication Technologies carry tremendous accessibility hope. Tim Berners-Lee, who invented the Web, wanted since its start *"to make the Web's benefits available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability"*. For us this aim is not limited to the Web and covers widely digital information. Unfortunately the high development speed of ICT lets insufficient space for reflexion about their accessibility, and most actors focus on main user interfaces, which makes the digital divide grow.

Our research activity takes place in the study of computer models allowing to use these technologies in order to improve access for all in the digital society, and in the integration of the principles of these specific models into mainstream models, which could be a way of defining "digital accessibility".

Nowadays electronic documents are now planned to be used within multimodal interfaces, but it is often interfaces specific to the application and not to the user, who has to adapt to this interface (the most common of course being screen/keyboard/mouse). Using alternative non-visual modalities such as Braille, speech synthesis or tactile interfaces requires a conversion that is often difficult to perform, particularly due to the fact that in mainstream models of representation, data are too often formatted exclusively for the main modalities used.

In order to fully exploit the potential offered by non-visual interfaces, we are working on models of representation that are independent of modalities, and which allow to adapt the

interface to the user's needs, and particularly to his/her visual capabilities. Visual impairment can have multiple forms and can be associated with various additional problems (cognitive, communication, motor, auditory). However non visual modalities have two common points, they are very linear, and they are unable to provide a global perception. Our goal is to develop tools that adapt automatically to the user, according to his/her specificity. First of course depending on the modalities the user can access, but also in according to the user's skills, particularly in the case of children. This adaptability is necessary in the case of educational use : adaptation to the user's progress, both in the school disciplines and in the use of tools. It is even more important in the case of an educational use with very young children or children with additional difficulties (cognitive, communication), who it is important to help into a progress process.

The works presented in this memoir thus refer to digital accessibility, and in particular accessibility to content that we call complex because they are structured and/or composite, having of elements of different natures. The application areas we have been studying are Websites, mathematical expressions and video games. After an introduction and presentation of the scientific context, in which we try to describe a number of practical problems relating to access to complex content, and the state of the art in this field, we present a selection of our works organised in four complementary parts, each corresponding to one or more of our application areas. The first part concerns specific user interfaces, allowing to make the most out of the specific modalities used by the visually impaired. The second part focuses on transmodal collaborative tools, that are tools allowing collaboration between people using different modalities around the same content, in order to exchange, work or play together. The third part describes transcription tools that we developed for converting the content formatted for a given modality into alternative modalities. Finally in the fourth part, we present a reflection on accessible models of representation, that are models containing the information required for presenting content according to various modalities, thus able to operate specific and/or collaborative interfaces presented in the first two parts, when necessary using the transcription tools presented in the third part. This study should be conducted in conjunction with current works on standards for representation of multimedia scenarios and school documents. In the field of video games we try to extend the results obtained in the TiM project about the study of gaming interaction and their adaptation to non-visual modalities, in order to propose a model of accessibility framework for creating games accessible to all ; that are games that users of alternative modalities can use, even using specific game interfaces allowing to access the original game and control it effectively.

The memoir concludes with a short chapter entitled "Accessibility and Society" in which we discuss accessibility in French society today, and strengths and limitations of technological tools to help the idea of accessibility to progress. In all areas, Accessibility not only helps to improve the lives of people with disabilities, allowing them to perform independently acts which before would necessitate permanent support, but also to improve the lives each person, by a better adaptation of tools to people and available resources. It is obvious that the availability of such tools is not enough to make society more accessible, but they are necessary and we believe they may contribute.

Table des matières

Avant-propos

Ce document constitue un complément au mémoire de synthèse présenté pour l'obtention de l'Habilitation à Diriger les Recherches. Il regroupe une sélection d'une quinzaine d'articles que nous avons publié, avec des collègues et des étudiants, au cours des 5 dernières années. Ils illustrent les travaux que nous avons mené depuis une douzaine d'années dans le domaine des interfaces humain-machine non visuelles et de l'accessibilité. Si dans le mémoire j'ai choisi une structuration transversale me permettant de relier ensemble des aspects similaires illustrés dans des domaines d'application différents, chaque article est par contre très lié au domaine d'application concerné. C'est pourquoi la structuration de ce document est plutôt faite dans ce sens, à l'exception du premier chapitre qui regroupe des articles de type *État de l'Art*.

Les articles sont regroupés selon 5 chapitres repartis comme suit. Des renvois vers le mémoire permettent de relier le contenu de ce dernier avec les articles.

- État de l'art
- MaWEn : aide à la compréhension d'expressions mathématiques et aux calculs
- Transcription : outils de transcription, des expressions mathématiques vers le Braille mathématique, ou plus généralement des documents de type texte (contenant éventuellement des mathématiques) vers des formats accessible
- Interfaces spécifiques de jeux multimédia : plusieurs études concernant l'utilisation d'une interface particulière pour présenter des jeux vidéos de façon accessibles.
- Accessibilité des jeux vidéo : quelles sont les conditions permettant à un jeu d'être accessible, comment progresser dans ce domaine vers une mise en accessibilité de plus en plus de jeux grand public

Afin de ne pas alourdir l'ensemble, nous avons choisi de ne pas présenter ici un certain nombre de sujets annexes sur lesquels nous avons travaillé. Il s'agit en particulier de la librairie libbraille[1] que nous avons développé dans le cadre du projet TiM, et qui est disponible en *Open Source* ; d'une brève étude sur le modèle *Open Source* et son application dans le domaine particulier des techniques d'assistance aux personnes ayant des besoins particuliers ; de la mise au point par l'association Les Doigts Qui Rêvent[2], qui édite des livres tactiles pour les enfants déficients visuels, d'un nouveau procédé de production de dessin en relief, le procédé ToM's 3D[3], auquel nous avons contribué dans le cadre du projet TiM ; ainsi que de travaux plus anciens sur l'accessibilité du Web, et dont on trouvera les références dans le document principal.

De même étant donné ma reconversion thématique vers les technologies d'assistance aux personnes, mes travaux antérieurs à 1997, concernant réseaux neuro-mimétiques pour des traitements du langage naturel ne sont pas présentés dans ce document ni dans le mémoire principal.

Dominique Archambault
Paris, novembre 2010

1 `http://libbraille.org`

2 `http://www.ldqr.org`

3 `http://www.toms3d.com`

Chapitre 1

État de l'Art

À l'occasion d'un numéro spécial de la revue en ligne *Upgrade*[4], intitulé *"Information Technologies for Visually Impaired"*, nous avons publié deux articles de type *État de l'Art*. Le premier concerne l'accès aux mathématiques pour les personnes handicapées de l'écrit. Nous y décrivons les modalités utilisables par les personnes handicapées visuelles – et en particulier nous y décrivons les différents codes mathématiques braille, les problèmes rencontrés, puis l'essentiel des projets de recherche ayant été menés sur ce sujet depuis une quinzaine d'années, et les outils existants les plus intéressants. Les articles cités sont classés par catégories : transcription en Braille, synthèse vocale, outils de navigation/aide à la compréhension des expressions, outils d'éditions. Nous concluons en évoquant des axes de recherche qui nous semblent devoir être explorés maintenant.

Le second traite de l'accessibilité des jeux vidéo. Dans cet article nous définissons la notion d'accessibilité des jeux vidéo, puis nous présentons une catégorisation des jeux spécifiques accessibles, c'est-à-dire des jeux développés spécifiquement pour un public handicapé. Ensuite nous abordons le domaine du *Design for All*, et citons les quelques projets ayant été menés pour développer des jeux accessibles à tous, personnes handicapées comme valides. Nous constatons que le besoin d'accessibilité croît dans nos sociétés, et nous présentons plusieurs initiatives dans ce domaine, comme la publication d'un livre blanc sur l'accessibilité par l'*International Game Developper Association*, et la constitution d'une collection de recommandations sur ce sujet. Pour terminer nous présentons notre positon dans ce domaine, qui est la base de plusieurs propositions de projets récents : la création d'une interface d'accessibilité spécifique aux jeux, et permettant aux dispositifs d'assistance d'accéder aux informations dont ils ont besoin sur le jeu en cours.

[4] `http://www.upgrade-cepis.org/`

1.1 [Upgrade 2007a]

Titre	**Access to Scientific Content by Visually Impaired People**
Auteur(s)	Dominique Archambault, Bernhard Stöger[(a)], Dónal Fitzpatrick[(b)] et Klaus Miesenberger[(a)]
Affiliations	[(a)]Johannes Kepler Universität Linz, Autriche [(b)]Dublin City University, Irlande
Type	Revue Internationale, avec Comité de lecture
Publication	Upgrade, Digital journal of CEPIS `http ://www.cepis.org/upgrade`
Editeur(s)	Josep Lladós-Canet, Jaime López-Krahe et Dominique Archambault *(Guest Editors)*
Volume/Date	Vol. VIII (2), Avril 2007
Pages	29–42

Cet article présente un état de l'art relativement exhaustif de ce qui se fait dans le domaine de l'accès aux Mathématiques pour les personnes handicapées de l'écrit. Il a été publié dans Upgrade, la revue en ligne du CEPIS. Une monographie en espagnol a été publiée dans la revue Novática.

Voir mémoire section **2.3.3**, page 20

Access to Scientific Content by Visually Impaired People

Dominique Archambault, Bernhard Stöger Donal Fitzpatrick , and Klaus Miesenberger

The study of Mathematics and Sciences has always been difficult for visually impaired students. In this paper we will describe the research undertaken during the past 20 years to support scientific work for blind and partially sighted people. We will first describe the modalities that can be used to render mathematical contents, and describe speech and Braille solutions, together with the inadequacies of these solutions. Then we will present a number of research projects organised in 3 categories: conversion based on Braille, conversions based on speech and navigation, and communication tools. We will then propose our views on the future research that needs to be carried out now, focusing on support functions for understanding and editing ("doing Maths"), and on communication between sighted and visually impaired people.

Keywords: Accessibility, Mathematics, Mathematical Braille, Speech, Visually Impaired People.

1 Introduction

The study of Mathematics has always been particularly difficult for blind people and especially for pupils in early classes who have to learn its specific notation. This is also the case for students who have to deal with very complex mathematical content. Most mathematical concepts are better explained using drawings and notes which illustrate the main content. These include graphics such as curves or geometrical figures, graphical notes (strokes, underlines or surrounding circles highlighting some parts of the material links between terms as illustrated by Figure 1, or textual material related to a specific part of the content. Additionally the mathematical notation itself uses two dimensions in order to convey more rapidly the general structure of the formula, which makes it easier to understand its semantic. One "pictures" the basic mathematical content at a glance, which helps to read the details in a more efficient way, since the role of every part of the expression is already assimilated.

When the visual modalities are not available, it is another story, indeed the other communication channels that are available to convey Mathematical contents (audio and tactile) do not offer the same possibility of getting a rapid overview. That makes it much more difficult for blind people to learn mathematical concepts than for sighted people. We can observe that a large majority of blind pupils do not succeed in Maths studies. To date in the Republic of Ireland, no totally blind student has been in a position to complete the Leaving Certificate (pre-university examination) at the higher level, and further evidence [1] demonstrates this fact. Clearly we assume that there is no reason that mathematical semantics can not be understood for reasons

Authors

Dominique Archambault is PhD in Computer Sciences and an Associate Professor in Université Pierre et Marie Curie, Paris. Since 1996 he works in the field of assistive systems for visually impaired people. He focuses on projects about non visual interfaces, Web accessibility and educational tools for blind children. One of his main topics of interest is the way the computer technologies can be used as a tool for the children's development, in particular in the case of visually impaired children, with additional difficulties (troubles coming from their visual impairment or additional disabilities). He was the co-ordinator of 2 IST Europeans projects (TIM - IST-2000-25298 and Vickie - IST-2001-32678). <dominique.archambault@upmc.fr>.

Bernhard Stöger, after studying Mathematics at the University of Linz, has been working as a researcher in the area of Functional Analysis in the Institute for Mathematics at the same university. Since 1991, he has been a member of the Institute for Integrated Study. His research activities here are centered about User Interfaces for blind people, focussing on Access to Mathematics. <bernhard.stoeger@jku.at>.

Dónal Fitzpatrick is a Researcher and Lecturer in computer science with particular interest in assistive technology at the School of Computing, Dublin City University. His PhD thesis focussed on transforming technical documents into synthetic speech and Braille. In 2000, he was awarded a fulbright scholarship to do a postdoc at Florida State University. Dr. Fitzpatrick's main interests are in the conveyance of highly visual material in a multi-modal fashion. He is presently engaged on the Socrates Minerva AHVITED project which is concerned with producing multi-modal renderings of diagrams in an e-learning environment. <dfitzpat@computing.dcu.ie>.

Klaus Miesenberger is the managing director of the Institute "integrated study" at University of Linz. After being guest professor at the *Université Claude Bernard, Lyon II*, he got his professorship (habilitation) in Human-Computer Interaction for People with Disabilities in 2001. He has been responsible for and has been working in more than 25 national and international research, development and application projects in AT and ICT for people with disabilities. He is responsible for the ICCHP (*International Conference on Computers Helping People with Special Needs*) since 1999. <klaus.miesenberger@jku.at>.

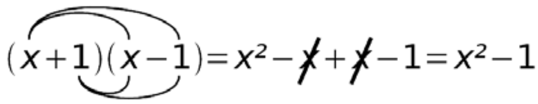


Figure 1: Extra Graphics Used to Illustrate a Calculation.

of blindness, rather the biggest barrier is the notation used for doing Mathematics. As this notation is essentially visual, it is quite difficult to transmit its contents using other modalities. Blind pupils often fail in Maths because of the way information is presented. A key feature which is present in the visual reading process is the role of the printed page. This medium affords the reader not only the facility to act as an external memory, but also facilitates a highly refined control over the flow of information. In the case of graphical content, tactile drawings can also be made. To some extent, this applies fairly to representation of curves and bidimensional geometrical figures. Even if great progress has been made in the technologies that allow the production of tactile drawings, there are limitations due to properties of tactile sense. It is also possible to render a curve using audio, by modelling a sound according to the shape of the curve, or using an haptic device like the Senseable Phantom [2].

In this paper we will focus on methods of access to mathematical formulas, which are the basis for all calculations and are used in all areas of mathematics and more generally in the sciences. Considering the two communications channels that have been cited above, formulas can be represented is a tactile form, usually based on Braille, or they can be spoken.

1.1 How People Read Mathematics

One of the key decisions which must be made when considering the manner in which mathematics, originally prepared using the visual modality, is depicted in either an audio or tactile one, is to firstly ascertain what information to present, followed by how to present this material. It is therefore important to understand the reading process, in order to fulfil the dual purpose of determining both **what** and **how** to present the relevant information to the user. In the subsequent paragraphs, the discussion is placed in terms of both auditory and Braille reading. and where there are discrepancies they will be highlighted.

A feature which is present in the visual reading process is the role of the printed page. This medium affords the reader not only the facility to act as an external memory, but also facilitates a highly refined control over the flow of

$$\frac{x + 1}{x - 1} \qquad x + 1 \ / \ (x - 1)$$

Figure 2: Linearisation of a Very Simple Fraction.

information. In his Ph.D. thesis, Stevens states that Rayner [3] describes reading as: "... the ability to extract visual information from the page and comprehend the meaning of the text" [3]. Stevens [4] also tells us that reading can be divided into three main domains.

1. The input of information from a physical, external source, into the reader's memory via the visual system;
2. The recognition of words and their integration into higher level structures such as sentences;
3. The process of understanding what has been read.

It would appear that there exists a point at which the process of listening and reading converge. This would seem to indicate that, once the information has been absorbed by the reader, it is both syntactically and semantically decomposed in the same manner, though the processes of actually retaining the material are quite different depending on which means the reader uses to read. It would appear that many readers hear a voice inside their head whilst reading. This voice articulates what is being read, giving the reader both a phonological and sub-localised impression of the document.

Stevens [4] defines the *phonological code* as "the auditory image kept in working memory during reading". It can be said that the written text is converted to this phonological code, which contains all the features of natural speech, such as pitch, rhythm etc. The notion of *inner speech* is quite speculative, but Rayner states that "Some proponents of inner speech have argued that reading is little more than speech made visible" [3]. The above appears to suggest that the visual component of reading is converted to an audio version, seeming to suggest a point where the two converge. After this point, the comprehension of the information should be the same. It is clear that the only differences in the reading process are the mechanical means of obtaining the information.

One aspect in which listening and reading differ significantly is the role of paper as an external memory. The manner in which the eye can relate to this external memory is a very powerful tool to aid in the retention and comprehension of written information. It can rapidly scan over the printed words, and by virtue of the juxtaposition of characters or symbols on the printed page, semantically interpret those symbols to produce the underlying meaning. Once the information is lost from the short term memory, it can be easily refreshed by the rapid movements of the eye.

There are a number of steps involved in the visual reading of a document. A skilled reader will normally read at a rate of 250-300 words per minute. The eye does not actually start at the top of a page and continue in a linear fashion until the end of the material is reached; rather the reading process consists of several distinct movements. Stevens tells us that there are a number of tasks which the eye performs in order to gain informational input. The reading process can be broken down into a series of **sacades** (jumps) and **fixations**. He tells us that:

"The sacades move the point of fixation in accordance with how much information has been or can be apprehended.

Figure 3: The *F*ormula (1) *R*endered *U*sing DotsPlus *T*echnology[1].

Forty nine percent of reading time is taken up with fixations. The rest of the time is taken up with the selection of which portion of the text to next fixate and the move to that location." [4].

In the Braille representation, while the *refreshable Braille Display* or printed page acts as an external memory, it is not possible for the finger to rapidly regress over material and assimilate it.

It is the absence of this external memory that is so different when the information is being read using audio. The facility of the paper as an external memory source is not present, and as the speech signal is transitory, the listener cannot easily recapitulate over already read material. Further, a vastly increased load is placed on the short term memory, thereby detracting from the ability to comprehend the material which could otherwise be easily understood.

An important distinction which must be made between reading and listening is the method by which the information is assimilated by the reader. When listening, the speech signal flows past the reader [4] [5]. The process can therefore be described as a serial exchange of information from the audio device to the passive listener. Visual reading, on the other hand involves rapid movements of the eye over a visual stimulus, and the acquisition of information through the visual cues which are inherently part of any document. Even the most rudimentarily written document consists of various forms of visual cue.

Firstly, the use of white space can determine the ends of both sentences, and other sectional units (such as paragraphs). This use of visual enhancement becomes more pronounced as the complexity of the material being presented increases. Finally, when a point is reached when mathematical (or other such technical information) is included, the use of the innate visual cues becomes more and more important to aid the visual reader in distinguishing the diverse types of content from the main body of the material. Consequently, material which is important can be easily recognised by virtue of the visual characteristics which have been imparted to it. The listener on the other hand is reliant on the prosodic features which form a part of all spoken output, while the Braille reader relies on a highly structured linear presentation in order to apprehend the information.

$$x+\frac{1}{x-1}\;(3)\qquad x+\frac{1}{x}-1\;(4)\qquad \frac{x+1}{x}-1\;(5)$$

Figura 4: Problems of Ambiguity when Speaking a Formula.

1.2 Linearisation

In both tactile or speech cases, the non visual representation necessitates linearising the formula. Any formula can be linearised, in most cases this linearisation generates a much longer representation, which is more difficult to understand than the graphical one.

For instance the very simple fraction in Figure 2 is usually written with a visual layout (1) that allows a very quick understanding of its semantics. The reader understands at first glance that the formula is a fraction and that the numerator and the denominator are rather short. The same formula written in a linear way (2) needs a bit more thought to be understood. Additionally it is written using many more symbols (11 while the first one only necessitates 7).

1.3 Tactile Representations

From this linear representation it is possible to print a Braille version using specific Braille characters for mathematical symbols (for example the square root). As Braille is based on a six-dot pattern, it only allow 64 different symbols, which is not enough to represent the large variety of Maths symbols necessary. Consequently most symbols are represented using several Braille characters, which makes the formulas even longer. For instance the formula in Figure 3 requires up to 19 Braille characters, depending on the Braille code used (see below). Note that this example is quite simple but in the case of complex formulas the number of symbols increases dramatically.

During the twentieth century, various strategies have been developed to reduce the length of formulas, which led to the design of specific Braille notations for Maths. These specific notations are usually context sensitive, which allows large reductions in the length of the formulas. These notations will be described further in the section 2 below.

There exist some representations based on an eight-dot Braille pattern: the Stuttgart Maths Braille Notation[6], which is used locally in Germany, and the lambda-code, designed by the EU-sponsored project of the same name [7], that will be described below. The advantage of eight-dot Braille is that 256 different characters can be represented, and therefore the length of formulas can be reduced even further. Unfortunately eight-dot Braille cannot be read by a simple horizontal motion of the finger. Indeed the surface-area of sensitive skin on the end of the finger is just sufficient to apprehend the six-dot pattern (except maybe in exceptional cases). Then it is necessary for the reader to move the finger vertically on each character, which makes the reading process very slow, so the benefit of reducing the length of the formula is lost.

Another tactile representation which deserves mentioned here, it is the so-called "dots plus"[1] developed by View

[1] <http://www.viewplus.com/products/braille-math/dotsplus/>.

Plus[8]. Using a "graphical" embosser the formulas are represented using a mix of Braille characters and graphics, in a way which is close to the graphical layout. Braille characters are used to represent the basic elements, like digits and letters, while the mathematical symbols are represented graphically: fraction bar, square root, arithmetic operators.

Additionally this representation keeps the graphical layout of the formula (for instance a fraction is represented by a graphical line with the numerator above and the denominator below).

This representation is quite interesting to help pupils to understand the meaning of a formula, but unfortunately it is based on paper printouts and therefore can only be used to read, rather than to write or edit a formula.

1.4 Speech

Speaking a formula is the other option. The linear version of the formula is said by a human speaker or by a speech synthesiser. In this case there are some non trivial problems of ambiguity. For instance if we consider the formula 1, it will be naturally spoken like the following sentence: *"x plus 1 over x minus 1''*. However, there are 3 other ways in which this sentence can be understood: see formulas in Figure 4.

Another obstacle to comprehension of audio-based mathematical information is the increase of the mental work involved in the retention and comprehension of audio-based mathematical information. Indeed whereas the reader can use the printed page as the external memory and an aid to retention, the listener has only their memory of the spoken utterance, thus making the comprehension of syntactically rich data extremely difficult.

Much of the investigation into the intelligibility of synthetic speech has been carried out using lists of single words, separated by pauses [9]. It was demonstrated that when the length of the pause was reduced, the retention was degraded far below that of natural speech. Waterworth conjectures that the reason for this is that listeners are exhibiting a recency or primacy effect [9]. It is inferred that the listener's working memory is concerned with either analysing and interpreting the acoustic input, or rehearsing material which is already present.

Coupled with this, Pisoni [10] has shown that the comprehension of synthetic speech depends on the quality of the system, and varies over a wide range; from 95.5% in the case of natural speech, to 75% when poor quality synthetic speech was in use. It can be further inferred that the intelligibility of spoken output is determined by:

- Whether the spoken output is synthetic or natural.
- The quality of synthetic speech.
- The level of prosody contained in the spoken utterance.

The prosodic component of speech [11] is that set of features which lasts longer than a single speech sound. The term *prosody* can be traced back to ancient Greek where it was used to "refer to features of speech which were not indicated in orthography, specifically to the tone or melodic accent which characterised full words in ancient Greek'' [12]. The term *prosody* remained almost forgotten until the 1940s, when it was revived as an approach to the study of linguistic analysis.

Another major factor in the understanding of synthetic speech is the *fatigue effect* which is primarily brought about by the monotonous quality of synthetic speech. It has been found that the introduction of prosodic cues into spoken output has increased intelligibility significantly. One possible reason for this is the relieving effect that the inclusion of prosodic features, such as alterations in the pitch range, and changes in the rate introduce a rhythm more akin to natural speech, hence relieving the tedium of the monotonous voice.

This fact has major implications for the presentation of syntactically complex material such as mathematical equations. Three sets of rules are known to exist for the production of spoken mathematics. The first is provided by the *Confederation of Taped Information Suppliers* (COTIS), the second is a set of guidelines written by Larry Chang [13], and the third is devised by Abraham Nemeth [14]. These rules attempt to alleviate the problem of syntactically rich material through the addition of lexical cues, adding to the mental workload of the listener.

Also, both these sets of guidelines are aimed at the human reader. Consequently, they are flexible enough to permit the semantic interpretation of the material, or to read the various symbols as they occur. In addition, the fact that both sets of rules are intended for human use assumes that the human reader can employ all features of natural speech when speaking the material. Such semantic interpretation is not available to any automated system; necessitating the development of a tighter set of rules to unambiguously present the material.

1.5 Computer Tools

During the past 3 decades, considerable progress has been made in the field of access to information for the group of blind and visually impaired people. Thanks to modern information technology in the mainstream and to very specialised adaptive and assistive technologies, blind and visually impaired people are now able to deal independently and efficiently with almost every piece of information that is composed of pure text.

Despite current strong trends towards graphical presentation, text still covers the majority of relevant contents for private and professional life, such that information access for this target group is currently accomplished to a very large extent.

On the other hand, blind and visually impaired people are still excluded from an efficient usage and handling of graphical content. Since Mathematics is presented in a highly graphical way most of the time, this exclusion implies considerable restrictions in access to Mathematics.

The problems faced by the target group with respect to Mathematics fall into four basic categories [15]:

1. Access to mathematical literature (books, teaching materials, papers etc.).

2. Preparation of mathematical information (presenting school exercises, writing papers etc.).

3. Navigation in mathematical expressions and communication between blind and sighted people.

4. Doing Mathematics (carrying out calculations and computations at all levels, doing formal manipulation, solving exercises).

We will present in Section 3 various published works which address the first two problems at least in a satisfactory manner (to some extent). In Section 7 we will focus on a few approaches that are now addressing the third one. Unfortunately until now almost nothing has been achieved to support the target group in solving tasks relating to the last category [16] [17] [18].

1.6 iGroupUMA

To address these challenges, 6 organisations having expertise in the field of Mathematics for the Blind have decided to join their efforts, creating the *International Group for Universal Math Accessibility* (iGroupUMA). They have been since joined by a seventh organisation.

The iGroupUMA members are:

- University of Texas at Dallas, United States.
- Dublin City University, Ireland.
- University of South Florida at Lakeland, United States.
- Johannes Kepler University of Linz, Austria.
- New Mexico State University, United States.
- University Pierre et Marie Curie in Paris, France.
- University of Kyushu, Japan.

2 Braille Mathematical Notations (BMNs)

2.1 General idea for BMNs

Braille is a linear writing system and consequently, it is necessary to linearise formulas to represent them in Braille. We have seen in Section 4 that the first "natural" way of writing formulas is to translate them to a non specific linear form, and then to use specific Braille characters for mathematical symbols, however this method makes formulas very long and quite difficult to handle for blind students.

In order to reduce the length of these formulas as far as possible specific Braille notations for Mathematics, and more widely for Scientific content, have been developed during the second half of the twentieth century. These very high level notations have been designed in order to improve the readability for the blind, mainly by significantly reducing the number of symbols necessary to represent a formula. To achieve this brevity, they have been based on context sensitive grammars which allow the use of the same symbol strings with different meaning depending on the context.

In counterpart these notations are quite difficult to learn (and to teach). The reason is that blind pupils have to deal with 2 learning difficulties at the same time: the Mathematical content itself and the Math code which is at least as difficult as the content. Currently only very proficient Braille users are able to do it, while average sighted pupils succeed much easier.

To further complicate things, these Braille Mathematical notations have been developed in different areas, according to the linguistic and cultural history of these countries. Therefore, while the mainstream (visual) representation of formulas is identical in every language, the same is not true for Braille notations. Indeed each Braille mathematical notation is widely used in its zone of linguistic influence, while it is completely unknown in other areas. In other words, a Braille formula written using the British notation is not understandable by a German speaking reader. This problem is quite important since the number of available Braille documents is very small compared to the number of ordinary Maths books.

The main Braille Mathematical notations are the following:

- In France, the Braille Mathematical code was first adapted to Mathematics in 1922 by Louis Antoine. This code was revised a first time in 1971. It was then deeply revised in 2001[19], in the goal of improving the collaboration between sighted and blind and facilitating automatic transcription. Nevertheless a lot of Braille readers still use the version devised in 1971.
- Marburg is used in German speaking countries. It was designed in 1955 in the Marburg school for the Blind in Germany by Helmut Epheser, Karl Britz and Friedrich Mittelsten Scheid. A heavily reworked and revised edition was published in 1986 [20].
- The Nemeth Code for Braille Mathematics was published and accepted as the standard code for representing math and science expressions in Braille in 1952. It was designed in 1946 by Abraham Nemeth so that he could complete his PhD in mathematics. The 1972 revision [21] is the current official code in use in the US. Note that Nemeth was adopted officially in a number of Southeast Asian countries (like India, Thailand, Malaysia, Indonesia, Cambodia, Vietnam).
- The British notation [22] is used in United Kingdom and in Ireland. It was first designed in 1970, and a deeply revised version was published in 1987. This was slightly revised in 2005.
- The current Japanese Mathematical Braille notation was published in 2001 by the Japan Braille Committee. It is an important revision of the 1981 formal specification of Japan Mathematical Notation, itself based on the notation published in 1956 by Japan Braille Research Group ("Nihon Tenji Kenkyukai”).
- Italy, Spain and many other countries have developed their own Mathematical notations.

Additionally some countries where no specific notations had been designed decided to officially adopt one of these notations. For instance Greece is using the Nemeth notation.

Finally a set of countries does not use such specific notations, like the European Nordic countries, but they use simple linearisation of formulas, with a set of specific symbols for Mathematical symbols that do not exist in the ordinary alphabet.

Figure 5: Transcription of Formula (1) in several Braille Mathematical Notations.

2.2 Main Strategies

Let us consider the transcription of formula (1) in Figure 2 in several Braille Mathematical notations (see Figure 5).

The number of available Braille symbols is quite reduced: 6 dots which can be combined in a maximum of 64 different patterns. Therefore it is necessary to use multiple Braille characters to code most Mathematical symbols. The various Braille notations implement different strategies for dealing with that problem. In Italian, the digits and the letters are always written using 2 symbols, the first indicating whether it is a digit or a letter and in the latter case which kind of letter (in the formula (8) for instance, "1" is

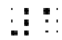
the first symbol standing for digit, while "x" is

where the first symbol means it is a Roman letter). British uses the same rule, but users will omit the prefix in the case of a symbol which cannot be a number (numbers are represented by the 10 first letters), like ‘x’ here

().

In Marburg the prefix for Latin letters is used only the first time, like a switch, indicating that any other instance of ‘x’ in the formula is of the same type (lower case Roman for instance here). Digits are always preceded by the

symbol. Finally in French the most frequent case is always assumed (lower case Roman), and there is a prefix before each other (upper case, Greek, etc.). There is also a special way to represent digits adding the dot ‘6’ to the corresponding letter traditionally used: instead of



the single symbol

is used (this is called the 'Antoine' notation for digits). This make simple formulas shorter. Nemeth differentiates also the digits from letters by using different Braille patterns. Here the numbers that are written in the lower part of the Braille cell: 1 is represented by

 .

Let us now consider the fraction itself. Block markers identify the numerator and the denominator. In French, Marburg and British notations the blocks are always the same; making it necessary to reach the fraction symbol to determine that this is in fact a fraction. The fraction structure itself uses 5 symbols, 2 pairs of block markers and a fraction mark (for example in French:

).

On the contrary in Italian the numerator and the denominator markers are not the same; there is no fraction symbol and the fraction structure uses only 4 symbols

and additionally the reader knows he/she is reading a fraction immediately from the first symbol. In the same kind of idea, Nemeth uses 3 Braille characters: the beginning of fraction, the fraction bar and the end of fraction

In order to reduce the length of simple numerical fractions, in Marburg and British notations, the denominator is written with lower numbers, that is, numbers that are written in the lower part of the Braille cell. For instance 1/4 will be

3 Conversion to/from Mathematical Braille

In the last twenty years, various projects have been developed in the field of Mathematical Braille notations, mainly with the aim of facilitating the written communication between sighted people and Braille readers in the field of Mathematics. We will first focus on converters that allow conversions to be performed between mainstream mathematical formats like LATEX and MathML and Braille notations.

These converters are used for different purposes. One is to facilitate the production of scientific Braille documents. Indeed it is much easier to produce a Mathematics document in LATEX or to use a word processor that supports MathML than to write a document in Mathematical Braille. Additionally a lot of resources are available in both these formats. In the reverse conversion (from Braille notations to mainstream formats) they allow sighted teachers or peers to access to formulas written by blind students.

3.1 Labradoor

Labradoor (*LAtex to BRAille DOOR*) converts a full LATEX document including Mathematical formulas into Marburg Braille or into HRTEX(see below). In addition, it offers a rich variety of formatting capabilities, enabling the production of Braille hard copies out of formatted LATEX documents [23]. As for conversion, one may choose between two options: The mathematical contents of a LATEX document may be converted either to Marburg Braille Notation, or to *Human Readable TeX* (HRTEX). The latter is a code developed at the University of Linz, with the intention to supply teaching materials in a way more easily readable than TEX or LATEX notation.

HRTEX is derived from TEX, although not compatible with it. These are some of the most important differences:

- Many symbols are abbreviated. For example, the symbols for Greek letters are composed of the first two characters, e.g., instead of `\alpha` we just write `\al`, instead of `\beta` we write `\be`, etc.
- The names of standard functions are written like variables, but in upper case letters, e.g., we write SIN instead of `\sin`, `LOG` instead of `\log`, etc.
- Alternative notation for fractions: The fraction bar is represented by two slashes `-//-`, and the whole fraction is written as a group. For instance, instead of `\frac{a+b}{c+d}` we write `{a+b // c+d}`.

As for formatting, Labradoor supports two output modes, one for Braille hard copies, and one for electronic text. In Hard Copy mode, elaborate text formatting algorithms suitable to represent paragraphs, lists, tables etc. in an attractive Braille layout are available. The table formatter deserves special mention, because it is able to represent tables in a variety of ways. Apart from the attempt to render a table in Braille according to its natural layout, tables may be dissolved according to their rows or columns. In Electronic Text mode these sophisticated formatting tools are disabled.

3.2 MathML to Braille Converters

Various converters from MathML to Braille have recently been developed. They allow transcribers to design Mathematical content using mainstream Maths editors.

Bramanet [24] converts formulas from MathML to French Braille. It is an application based on the XSLT technology. It allows various settings including the possibility to edit the output Braille table in order to fit with any hardware. It comes with a script which automatically makes a conversion from a document containing Maths formulas to a document ready to be embossed in Braille.

math2braille [25] is a "self-contained Module'' which takes in a MathML file and outputs a Braille representation of the same Maths. It is based on protocols and procedures that have been developed in a previous project about access to music. It produces Braille code in use in the Netherlands.

Stanley and Karshmer [26] propose a translator from MathML to Nemeth Braille Code. The translation is performed in 2 phases. First the MathML elements are translated to Nemeth codes.

Then syntactic rules that are inherent to Nemeth code are applied, such as the use of the numeric indicator, additional spaces, and some contractions. These rules were fashioned by Dr Nemeth to direct the conversion of Mathematics into the Braille code.

3.3 Insight

Based on the MAVIS project [27] which was the first solution to back-translation from Nemeth Braille code to LATEX, the Insight project [28] proposes a complete system to translate Maths documents with mixed Grade II Braille text and Nemeth code to LATEX. The back-translator is based on language semantics and logic programming.

The system processes an image of a Braille sheet (for instance a scanned page) and recognises the Braille dots to produce an ASCII Braille file. Text and Nemeth code are automatically identified and separated to be separately translated. Finally a single LATEX document is produced to be read by a sighted individual.

3.4 MMBT

MMBT (*Multi-Language Mathematical Braille Translator*) [29] was an open-source project allowing transcriptions from and to LATEX, MathML, French (revisions 1971 and 2001), British and Italian Braille notations. MMBT has been developed in Java to be used within the Vickie user interface (see the Vickie project below). In order to permit transcription from and to any of the supported formats, MMBT was based on a specific central language. It was discontinued and replaced by UMCL which uses the standard MathML as central language.

3.5 Universal Maths Conversion Library (UMCL)

One problem that transcribers have to deal with is the fact that most conversion tools are attached to one specific Braille code. For instance a French transcriber who looks for a tool to convert LATEX documents to Braille will find Labradoor, but this will be useless since it produces only Marburg.

One of the goals of the iGroupUMA has been to produce a programming library encapsulating various converters for various Braille codes in a single library usable through a simple and unique API. This library will also be useful for transcription tools (from mainstream notations to Braille and vice versa) as well as for software that needs real-time conversions (like formula browsers or editors). It will also make it possible to convert a document from one Braille national notation to another, increasing de facto the number of documents available for students and allowing blind mathematicians from various countries to exchange documents.

The UMCL (*Universal Maths Conversion Library*) [30] is an open-source portable library which allows such conversions. It was developed in standard "C" and has wrappers to different programming languages.

To make this possible without increasing the complexity, it was necessary to adopt an architecture based on a central representation of the formula, and to develop parsers to generate this central representation from the different input formats, and output generators for each output format [15] [31] [32].

The iGroupUMA has decided to use MathML as the central language. The architecture of UMCL is quite simple. It includes a main module and as many input and output modules as Maths codes are available. Input and output modules are independent and can be installed later on. The main module detects the available input and output modules and calls the modules necessary to perform the conversions according to requests. In other words it is possible to add input or output modules to any application using UMCL, at any time when a module becomes available.

Most output modules (MathML to Braille) are developed as XSLT stylesheets. Note that this is not mandatory, since input or output modules can be developed in any language. Interfaces for input and for output modules have been published.

In order to simplify the development of output modules, and to speed-up the processing time, we have developed a subset of MathML, that we call Canonical MathML [33]. Canonical MathML is a tentative attempt to unify MathML structures in a deterministic way in order to simplify transcription into Braille. All Mathematical structures that are necessary to perform a correct transcription into Mathematical Braille are recognised and rewritten in a unique way. Additionally Canonical MathML is valid MathML so it can be used with common tools which handle MathML. An input module for MathML allows the conversion from any MathML document to Canonical MathML. Currently output modules have been developed for the French notations (revisions 1971 and 2001) and Italian. Beta versions of Marburg and British code are also already available.

4 About source documents

Whatever the quality of the converter, it has to be mentioned here that converters can only transform what they are given! This seems obvious but one main problem encountered by transcribers is the poor quality of sources. Indeed, most documents come with formulas split in parts. For instance, for some reason there is often a character in between that is not included in the Mathematical material, because it looks nicer that way. This is also the case with LATEX input and with documents designed with word processors including Equation objects.

Additionally documents should be properly structured. Arrabito [34] states that without a degree of semantic markup, the production of a TEX based Braille translator, and indeed a universal Braille translator, is impossible. He points out that using the TEX primitives, authors can control the visual modality of their document, with no regard for the overall structure of their material. They could, for example, define various characters or symbols in terms of line segments, or use environments to achieve display styles for which they were not intended. Fitzpatrick [11] confirms this conjecture. Using the TEX language, some authors often use the "display'' environment to display textual material, as opposed to the mathematical content for which it was designed.

In this respect LATEX is much better than TEX since it is based on semantic mark-up. When using a word processor, it is important to use styles to define the structure, since these styles can be processed.

5 Infty

Another project deserves to be highlighted here. Infty [35] is a large project aimed at giving access to printed mathematical content. It is base on a core module (InftyReader) which is an OCR specialising in Mathematical documents [36]. It is able to extract mathematical formulas from a scanned document and to recognise the mathematical structure and the symbols. It produces a topological representation of the formulas in an XML format. Then this representation can be converted into various formats: MathML, LATEX, HTML, HRTEX, KAMS, and into Unified Braille Code (English) and Japanese Braille code.

Some additional tools have been developed by the same group, in order to extend the possibility of the system. InftyEditor allows the output of InftyReader to be edited. It proposes also a handwriting dialog in which users can write formulas by hand, to be recognised automatically. ChattyInfty [37] is adding speech output to the system. It allows visually impaired users to access expressions with speech output, and to also author and edit them.

6 Conversion to Audio

6.1 ASTeR

One of the most important attempts to produce accessible technical documents to date is the ASTER system [5]. ASTER (Audio System for Technical Reading) aims to produce accurate renderings of documents marked up in the TEX family of languages. ASTeR uses both spoken and non-speech audio to assist in this process. The tool used by the ASTER system to produce the audio output, is a language devised by the author known as *Audio Formatting Language*, or AFL. This language, an extension to LISP can be described as "... the audio analogue of visual formatting languages such as Postscript" [5]. The aim of this language is to provide mechanisms to control the multiple aspects of the audio presentation, such as speech-based, and non-speech sounds. The output produced by the audio-formatter in ASTER represents the various components of the audio presentation using *audio space*, which is derived by taking the sum, or cross product of the various individual dimensions of the audio space. Examples of these dimensions would be the spoken utterance, and the earcons[2] which enhance the spoken output. The output from the *audio formatter* is altered by adjusting the dimensions (parameters which may be changed) of each individual aspect of the audio space. Such dimensions would include the pitch and rate of the voice, the means by which these characteristics change to reflect alterations in the visual appearance of the text, etc.

Raman has observed that there is little in the way of similarity between the evolution of a written mathematical notation, and the audio counterpart. He points out: "Any notational system is a combination of conventions and intuitive use of the various dimensions that are provided by the perceptual modality and the means available to produce appropriate output for that modality.".

Raman [5] also points out that the traditional mathematics notation uses a set of primitive layout cues to achieve highly complex and often nested structures. The main aim of ASTeR's audio representation of mathematics is to produce a non-visual counterpart to the highly complex visual writing of mathematics. The system used in this particular software approach is to offer the listener the option to obtain highly descriptive renderings of the mathematics, or conversely purely notational. The former can be used when new material is being perused, while the latter can be utilised when familiar content is encountered.

6.2 The Maths Project

MathTalk [4] is another example of a system designed to render algebra more accessible to blind students. Unlike the ASTeR system described in the previous section, this system is aimed at students who are still at school, and is particularly aimed at those between the ages of 16 and 18. Accordingly the form of presentation, and its interface, are deliberately simpler and more intuitive to use. The design principles on which this work are based attempt to render the algebraic formulae in a non-interpretive fashion. This contrasts totally with the approach taken in ASTeR, where the mathematical content was rendered to make it more intelligible. The MathTalk program uses a vastly different prosodic model to convey both the content and structure of algebraic formulae. Stevens uses the following list to summarise the general rules implemented in MathTalk for the provision of lexical cues.

- For Latin letters, MathTalk only prefixes uppercase with a tag.
- For sub-expressions, MathTalk uses only the tag "quantity".
- MathTalk speaks simple fractions (those with a single term in both numerator and denominator) with only the word "over" between the two terms.
- Complex fractions (those with more than one term in either numerator or denominator) are bounded with lexical cues.
- Roots are enclosed with the lexical tags "the root" and "end root".
- Simple roots are spoken without an end tag.
- Initially MathTalk used the cue "to the" to indicate exponents. Later this was replaced by "super" (shortened from "superscript") to comply with minimal interpretation.
- The word "all" can be used with the opening superscript cue, when the superscript governs a complex object [4].

6.3 The TechRead System

The TechRead system [38] [39] departed from the premises described in the preceding systems. This approach aimed to produce a rendering of mathematics without using non-speech audio; rather relying on the prosodic features found in synthetic speech to convey the material. The core idea involved the use of alterations in the pitch, speaking rate and amplitude of the voice to convey the content, and the introduction of strategically placed pauses to convey the structure.

The paradigm on which the spoken mathematical presentation is based [40], is that of converting a sequence of juxtaposed symbols, delimited by both white space and other visual cues (such as parentheses) into a serially transmitted linguistic approximation. In order to achieve this, a parallel was drawn between the structure found in mathematical expressions and the inherent composition of English sentences [11].

The two methods used to imply the grouping of terms into sub-expressions, (and by extension of sub-expressions into the whole formulae) is to insert pausing, and alter the

[2] audio equivalents of icons.

speaking rate at strategic points within the presentation. This is not an arbitrary process but is based on both the mathematical components preceding and following the point at which the pause is needed. For example, the expression

$$a + b + c$$

a simple, linear concatenation of three quantities, separated by two relational operators, irrespective of whether the material is being read visually or auditorily. However, the expression

$$\frac{a}{b} + c$$

or non-linear in the print medium, but linear in the audio domain.

Accordingly, something must be introduced to indicate to the listener that the fractional component of the expression merely encompasses the first two elements, and that the final one is not contained therein. The method whereby this is achieved, is to speak the expression as follows: "a over b, plus c"'. Using this method, it can be clearly seen that the distinction between the fractional component of the expression and the remaining relational operator and quantity is evidenced. If one also adds a slight quickening of the speaking rate of the utterance of the fraction, then the distinction becomes even more apparent to the listener.

The fundamental core of the prosodic model used in TechRead is that of relative rather than absolute change. The alterations in the pitch and speaking rate, and the duration of the pauses are all calculated in terms of a starting point rather than as fixed quantities. This facilitates use of the system by both experienced and novice users, and also enables a high degree of flexibility in the model used.

6.4 MathPlayer

MathPlayer is a plug-in for MS Internet Explorer[3] [41]. MathPlayer works with screen readers using MS Active Accessibility interface (MSAA). It actually sends to the screen reader a string containing the relevant sentence to read with a speech synthesiser. A MathZoom feature allows expressions to be magnified. Built-in speech rules try to minimise the number of words used to speak the math while producing an unambiguous rendering of the math.

MathPlayer ability to use prosody is severely limited by MSAA, which only allows a string to be sent to the screen reader. Nevertheless MathPlayer provides an interface for screen readers to get round this problem.

7 Navigation and Communication Tools

7.1 The Vickie Project

The Vickie Project[4] [42] aims at facilitating the inclusion of visually impaired pupils and students in mainstream education.

A unified user interface has been developed to enable visually impaired pupils and students to access documents using specific devices, while a synchronised graphical display supplies sighted teachers with a view of the same document. This interface is an application built on the framework of Mozilla[5]. The interface actually presents 3 views of the same document:

- A graphical view of the document, including images and mathematical formulas.
- This view is called the teacher's view.
- An adapted view to partially sighted users, with adjustable print and colours.
- A Braille view. The libbraille library [43] is used to write directly on Braille devices.

The mathematical formulas are stored in documents in the MathML standard, and then can be displayed easily in a graphical way thanks to Mozilla's MathML rendering engine. This rendering engine permits the setup of the size of print and the colours for the adapted view.

The formula is automatically converted into Braille in real-time thanks to the MMBT converter [29] (see above). This software will also convert formulas written by the pupil in Braille to MathML in order for it to be displayed for the teacher.

In the Vickie project all documents are stored on a server [44] using the dtbook DTD format[6]. The mathematical formulas are included in the dtbook documents in MathML, thanks to the module mechanism of the dtbook DTD. Then the MMBT is also used to convert documents from a specific Braille notation that is used in France for printing (BrailleStar). Numerous documents are stored in that format and can then be imported into the Vickie server.

7.2 Math Genie

The Math Genie [45] is a formula browser that facilitates understanding of formulas using voice output. It has recently been augmented to provide Braille output on refreshable Braille displays. It has been designed to convey the structure of the mathematical expression as well as its contents. It can be used by a blind student together with a sighted teacher who does not need to have any specific knowledge in Braille. Indeed the teaching material can be prepared for both sighted and blind students using any Math editor able to produce MathML. The graphical rendering is synchronised to the audio which makes communication easier with the teacher. It is based on SVG (Scalable Vector Graphics), which allow the Math Genie to support magnification, in order to give support to partially sighted individuals, and colour-contrasted highlighting, in order to support individuals with dyslexia. It should be remarked that this project is based on research in cognitive psychology, in

---

[3] MathPlayer is a free download from <http://www. dessci.com>.

[4] The Vickie Project is funded by the European Commission, on the IST (Information Society Technologies) Programme 2001 (FP5/IST/Systems and Services for the Citizen/Persons with special needs), under the reference IST-2001-32678.

[5] The Mozilla project: <http://www.mozilla.org>.

[6] The dtbook Document Type Definition (DTD) is part of the NISO Z39.86-2002 standard, also known as DAISY 3.0.

order to enhance the way voice synthesis reads mathematical formulas [46]. The Math Genie offers blind students several ways of reading the formulas, from default reading from left to right to an abstract way that highlights the hierarchical structure while "folding" away the sub-expressions. The user can navigate in the mathematical structure, moving by way of meaningful "chunks". This is based on lexical clues, which represent the structure of the mathematical content.

Additional features provided by the Math Genie are:

- the possibility to let the user to add so-called "voicemarks" to expressions, that is to record audio bookmarks attached to any sub-expression.
- an online dictionary of mathematical terms accessible during navigation in mathematical expressions through simple keyboard shortcuts gives contextual support to the blind users.

The current version supports English, French and Spanish for speech, and offers facilities to add any local language provided that a speech synthesiser is available with the requested interface. The Braille output currently supports the Nemeth code [26] (see Section 3.2).

7.3 Lambda

Lambda [6] is a mathematical reading and writing system designed for blind students. The software was developed in a project of the same name, whose meaning is in full: "Linear Access to Mathematics for Braille Device and Audio Synthesis". The Lambda software is mostly referred to as the "Lambda Editor''. It is an editor that enables a blind student to input and to edit mathematical expressions in a rather comfortable way.

The main characteristics of the Lambda project is that it is built on a new code. This code is an XML code specifically designed for supporting the Braille transcription into 8-dot pattern national codes. Each Lambda national code has the lambda structure and a Braille character dictionary as close as possible to the official national code.

Within Lambda, mathematical formulas may be input in several ways:

- through keyboard shortcuts.
- from a structured menu.
- from an alphabetic list.
- from a toolbar (for sighted users mainly.).

As for output, Lambda supports these modalities:

- Braille output in a special, though customisable 8 dot code.
- speech synthesis - mathematical symbols are verbalised in a descriptive language.
- visual presentation in a linear code (a specific font in which each Braille character is represented by a visual symbol).
- graphical rendering - not synchronous to input, the graphical rendering is built when the user presses a key. This view is then static. The graphical view is obtained by conversion of the Lambda code to MathML.

Lambda offers several tools to support a student in editing mathematical expressions. The most important among these utilities is the manipulation of blocks: Every structure with an opening and a closing tag, i.e., an expression in parentheses, a root, a fraction, an index, or an exponent, is considered a block. The Editor has functionality to select a block, to enlarge or diminish the selection according to structural levels, to delete a block, and to copy it to a different place in the document.

As for navigation support, Lambda offers collapse and expand functionality, which is also organised along the block structure of a mathematical expression.

8 Future Challenges

Nowadays, the significant efforts made around the MathML language and the progress made by rendering programs (like Mozilla for instance) and the equation editing software (like MathType) allow us to develop very useful software that might help blind users to deal with the intrinsic complexity of Mathematical Braille notations. We have seen in this paper a few examples that provide such help (Math Genie, Lambda).

8.1 Needs

One of the central objectives of works in this field is now collaborative work between blind and sighted individuals, most typically in a mainstream teaching environment, where one blind pupil needs to be able to collaborate with his/her sighted teacher and, perhaps, several sighted fellow students.

This requires synchronisation of 2 representations using 2 different modalities, one dedicated for the blind and one dedicated for the sighted. Each of these representation must be the natural representation, that is the representation the readers are used to. In the case of sighted people it has to be the natural graphical view. In the case of blind readers it has to be the official Braille notation in use in their environment.

The synchronisation must allow each one to point a location on the formula to show it to the other, in order to highlight an idea or to explain an error. On the graphical view this pointing must be done using the mouse by clicking on the desired location. Then the specified location is highlighted on the Braille display. On the other direction the Braille user can click on a cursor routing key and then make appear the selected location with a different background on the screen.

Additionally it is necessary to be able to collapse/expand some branches of the Maths expression in order to get overviews of the formulas. Obviously this has to be synchronised too.

Actually carrying out mathematical calculations is even more difficult than reading and writing formulas. The problems in doing formal manipulations happen because of the complex structures that arise during a calculation: whereas sighted people may organise a computation such that it can be easily surveyed and navigated, blind people tend to get lost quickly within a formal structure. There is a need for powerful editing tools that pupils can use to do calculations more easily.

There is also a tremendous need for tools providing contextual support on the Braille mathematical code and doing calculations. These tools should provide support with respect to the Braille notation itself and not mathematical content. The aim is to reduce the gap between blind pupils and their sighted peers induced by the complexity of mathematical Braille notations.

8.2 MAWEN

We are currently developing a prototype application based on MathML which implements all these features. MAWEN, which stands for "Mathematical Working Environment", is currently being developed in co-operation between the Johannes Kepler University of Linz, Austria, and the University Pierre et Marie Curie in Paris, within the MICOLE project, funded by the European Commission. It is a comprehensive, collaborative, bi-modal software solution designed to address all the basic problems outlined in this paper.

The system is designed:

- To work on documents of mixed content - textual and mathematical.
- To simultaneously represent formulas in a Braille Mathematics code of the user's choice (MAWEN potentially supports any official Braille code - as soon as it is implemented in the UMCL library), and in natural visual rendering.
- To support bi-directional pointing possibilities.
- To support navigation through formulae by collapse and expand functionality, synchronised with both views.
- To input/edit this mixed content, and especially mathematical formulas, such that the above-mentioned simultaneous representation persists.
- To support the student in doing mathematical manipulation.

In order to achieve this objective we have developed a model which is based on MathML - actually on Canonical MathML and which supports synchronisation and all features described above. The choice of a standard (MathML) as work representation ensures the timelessness of the system.

9 Conclusion

The study of Mathematics is crucial in most science disciplines. The difficulty inherent in the particular notation they use clearly disadvantages blind and partially sighted pupils. Therefore there is a urgent need for software tools which help them to overcome the difficulty due to their impairment. Considering the current trends which encourage more and more such pupils to attend mainstream schools it is necessary that these tools are usable by teachers who do not have a specific knowledge of Braille. Today the development of inexpensive but powerful computers allows to have an optimistic view on the future.

Indeed we have seen in this paper that a lot of partial solutions have been developed in the past decade. The involvement of multidisciplinary teams to have a better knowledge about the way individuals understand and represent Math content will help to make these tools more efficient and useful.

The development of tools based on MathML (which is now the standard for representing Mathematics contents) allows the development of tools that will better integrate visually impaired people into the mainstream. Indeed these tools allow, for instance, the production of documents with mainstream software. Additionally we can access software tools which allow high quality graphical rendering of MathML formulae. The development of converters from MathML to various Braille code in a portable and modular form will allow to integrate the natural representation of formulae for each user in an efficient work environments, giving some support on the math code to the blind users.

References

[1] K. A. Ferrell, M. Buettel, A. M. Sebald, R. Pearson. American printing house for the blind mathematics research analysis. Technical report, National Center on Low-Incidence Disabilities - University of Northern Colorado, 2006.

[2] W. Yu, S. Brewster. Evaluation of multimodal graphs for blind people. Journal of Universal Access in the Information Society, 2(2):105-124, 2003.

[3] K. Rayner, A. Pollatsek. The Psychology of Reading. Prentice Hall, 1994.

[4] R. D. Stevens. Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People. PhD thesis, Department of Computer Science, The University of York, Heslington, York, UK, 1996.

[5] T. V. Raman. Audio Systems for Technical Reading. PhD thesis, Department of Computer Science, Cornell University, NY, USA, 1994.

[6] W. Schweikhardt. *Stuttgarter mathematikschrift für blinde. Universität Stuttgart, Institut für Informatik*, 1998.

[7] W. Schweikhardt, C. Bernareggi, N. Jessel, B. Encelle, M. Gut, LAMBDA: A European System to Access Mathematics with Braille and Audio Synthesis. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), volume 4061 of LNCS, pp. 1223-1230, Linz, Austria. Springer, 2006.

[8] J. Gardner, C. Herden, G. Herden, C. Dreyer, G. Bulatova. Simultaneous braille tactile graphics and ink with tiger ink attachment and duxbury. In Proc. CSUN'04 (19th Annual Conference "Technology and Persons with Disabilities"), Los Angeles, California, USA, 2004.

[9] J. A. Waterworth. The psychology and technology of speech, symbiotic relationship. In J. A. Waterworth,

editor, Speech and Language Based Interaction with Machines: Towards the Conversational Computer. John Wiley, 1987.

[10] J. V. Ralsten, D. Pisoni, S. E. Livel, B. G. Green, J. B. Moulinix. Comprehension of synthetic speech produced by rule. Human Factors, 33(4):471-491, 1991.

[11] D. Fitzpatrick. Towards Accessible Technical Documents: Production of Speech and Braille Output form Formatted Documents. PhD thesis, School of Computer Applications, Dublin City University, 1999.

[12] E. Couper-Kuhlen. An Introduction To English Prosody. Edward Arnold Ltd, 1986.

[13] L. A. Chang. Handbook for Spoken Mathematics (Larry's Speakeasy). Lawrence Livermore Laboratory, The Regents of the University of California, 1983.

[14] A. Nemeth. Tips for Reading Math Out loud to Blind Students. <http://www.rit.edu/ easi/easisem/talkmath.htm>.

[15] M. Batusic, K. Miesenberger, B. Stöger. Parser for the marburg mathematical braille notation. In C. Stephanidis, editor, Universal Access in HCI - Inclusive Design in the Information Society, Vol. 4, pages 1260 - 1264, Mahwah, New Jersey, USA. Lea, 2003.

[16] H. Guo, A. Karshmer, C. Weaver, J. Mendez, S. Geiger. Computer processing of nemeth braille math code. In Vollmar, R. and Wagner, R., editors, Proc. ICCHP 2000 (International Conference on Computers Helping People with Special Needs), Universität Karlsruhe, Germany. Österreichische Computer Gesellschaft, 2000.

[17] A. Karshmer. Access to mathematics by blind students. In K. Miesenberger, J. Klaus, W. Zagler, editors, Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs), Vol. 2398 of LNCS, Linz, Austria. Springer, 2002.

[18] B. Stöger, K. Miesenberger, M. Batusic, D. Archambault. Multimodal interaction with mathematical formulae: Access, communication and support for blind people in doing mathematics. In Stephanidis, C., editor, Proceedings of HCI International 2005 Conference, Las Vegas, Nevada. 10 pp. (proceedings on CD-Rom).

[19] Commission Évolution du Braille Français. Notation mathématique braille. INJA and AVH, Paris, France, 2001.

[20] H. Epheser, K. Britz, F. M. Scheid. *Neufassung und vervollständigung des systems der internationalen mathematikschrift für blinde.* Marburg/L.: *Ständige Arbeitsgemeinschaft "Mathematikschrift", Deutsche Blindenstudienanstalt e.V.*, 140 pp.

[21] A. Nemeth. The nemeth braille code for mathematics and science notation, 1972 revision. American Printing House for the Blind, USA, 1972.

[22] Braille Autority of the United Kingdom. Braille mathematics notation. RNIB, Peterborough, UK, 2001.

[23] M. Batusic, K. Miesenberger, B. Stöger. Labradoor, a contribution to making mathematics accessible for the blind. In A. Edwards, A. Arato, W. Zagler, editors, Proc. ICCHP 98 (6th. International Conference on Computers Helping People with Special Needs), Oldenbourg, Wien, München.

[24] F. Schwebel. *BraMaNet logiciel de traduction des mathématiques en braille*, 2003. <http://handy.univ-lyon1.fr/projets/bramanet/>.

[25] D. Crombie, R. Lenoir, N. McKenzie, A. Barker. (2004). math2braille: Opening access to mathematics. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Vol. 3118 of LNCS, pp. 670-677, Berlin. Springer.

[26] P. B. Stanley, A. I. Karshmer. Translating mathml into nemeth braille code. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp. 1175-1182, Linz, Austria. Springer, 2006.

[27] A. I. Karshmer, G. Gupta, S. Geiiger, C. Weaver. Reading and writing mathematics: the mavis project. In International ACM Conference on Assistive Technologies (ASSETS), pp. 136-143, Marina del Rey, CA, USA, 1998.

[28] A. Annamalai, D. Gopal, G. Gupta, H. Guo, A. Karshmer. (2003). INSIGHT: a comprehensive system for converting Braille based mathematical documents to Latex. In C. Stephanidis, editor, Universal Access in HCI - Inclusive Design in the Information Society, Vol. 4, pp. 1245 - 1249, Mahwah, New Jersey, USA. Lea.

[29] V. Moço, D. Archambault. Automatic translator for mathematical braille. In C. Stephanidis, editor, Universal Access in HCI - Inclusive Design in the Information Society, Vol. 4, pp. 1335 - 1339, Mahwah, New Jersey, USA. Lea, 2003.

[30] D. Archambault, D. Fitzpatrick, G. Gupta, A. Karshmer, K. Miesenberger, E. Pontelli. (2004). Towards a universal maths conversion library. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Vol. 3118 of LNCS, pp. 664 - 669, Berlin. Springer.

[31] A. Karshmer, G. Gupta, K. Miesenberger, E. Pontelli, H.-F. Guo. The development of a tool to enhance communications between blind and sighted mathematicians, students and teachers: A global translation appliance. In C. Stephanidis, editor, Proceedings of the UAHCI'01 conference — Universal Access in Human-Computer Interaction (joint with 9th International Conference on Human-Computer Interaction) — Towards an Information Society for All, pages 929 - 932, New-Orleans, Louisiana, USA. Lawrence Erlbaum Associates, 2001.

[32] B. Palmer, E. Pontelli. Experiments in translating and navigating digital formats for Mathematics (a progress report). In C. Stephanidis, editor, Universal.

[33] D. Archambault, V. Moço. Canonical mathml to simplify conversion of mathml to braille mathematical notations. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp. 1191-1198, Linz, Austria. Springer.

[34] R. Arrabito. Computerized Braille Typesetting: Some Recomendations on Mark-up and Braille Standard. Technical report, The University of Western Ontario, 1990.

[35] M. Suzuki, T. Kanahori, N. Ohtake, K. Yamaguchi. An Integrated OCR Software for Mathematical Documents and Its Output with Accessibility. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Vol. 3118 of LNCS, pp. 648-655, Berlin. Springer, 2004.

[36] R. Fukuda, N. Ohtake, M. Suzuki. Optical recognition and braille transcription of mathematical documents. In R. Vollmar, R. Wagner, editors, Proc. ICCHP 2000 (International Conference on Computers Helping People with Special Needs), pages 711-718, Universität Karlsruhe, Germany. Österreichische Computer Gesellschaft.

[37] T. Komada, K. Yamagushi, F. Kawane, M. Suzuki. New Environement for Visually Disabled Students to Acess Scientific Information by Combining Speech Interface and Tactile Graphics. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp. 1183-1190, Linz, Austria. Springer, 2006.

[38] D. Fitzpatrick. Mathematics: How and What to Speak. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp. 1199-1206, Linz, Austria. Springer, 2006.

[39] D. Fitzpatrick, A. Karshmer. Multi-modal mathematics: Conveying math using synthetic speech and speech recognition. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Vol. 3118 of LNCS, pp. 644 - 647, Berlin. Springer.

[40] A. Monaghan. Intonation in a Text-to-Speech Conversion System. PhD thesis, University of Edinburgh, 1991. Access in HCI - Inclusive Design in the Information Society, Vol. 4, pp. 1245 - 1249, Mahwah, New Jersey, USA. Lea, 2003.

[41] N. Soiffer. Advances in accessible web-based mathematics. In Proc. CSUN'05 (20th Annual Conference "Technology and Persons with Disabilities"), Los Angeles, California, USA, 2005.

[42] D. Archambault, D. Burger. The vickie project. In K. Miesenberger, J. Klaus, W. Zagler, editors, Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs), Vol. 2398 of LNCS, pages 90 - 97, Linz, Austria. Springer.

[43] S. Sablé, D. Archambault. libbraille: a programming library for easy access to Braille displays. In G. Craddock, L. McCormack, R. Reilly, H. Knops, editors, Assistive Technology - Shaping the future, Proceedings of the AAATE'03 Conference, Dublin, Ireland, pp. 435 - 440, Amsterdam, The Netherlands. IOS Press. Schwebel, 2003.

[44] B. Guillon, J.-L. Monteiro, C. Chécoury, D. Archambault, D. Burger. Towards an integrated publishing chain for accessible multimodal documents. In J. Klaus, K. Miesenberger, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs), LNCS, Linz, Austria. Springer, 2004.

[45] A. Karshmer, G. Gupta, S. Geiger, C. Weaver. Reading and writing mathematics: The mavis project. Behavior and Information Technology, 18(1):2-10, 1998.

[46] A. Karshmer, D. Gillan. How well can we read equations to blind mathematics students: some answers from psychology. In Stephanidis, C., editor, Universal Access in HCI - Inclusive Design in the Information Society, Vol. 4, pp. 1290 - 1294, Mahwah, New Jersey, USA. Lea, 2003.

1.2 [Upgrade 2007b]

Titre	**Computer Games and Visually Impaired People**
Auteur(s)	Dominique Archambault, Roland Ossmann[(a)], Thomas Gaudy[(b)] et Klaus Miesenberger[(a)]
Affiliations	[(a)]Johannes Kepler Universität Linz, Autriche [(b)]CNAM, Paris
Type	Revue Internationale, avec Comité de lecture
Publication	Upgrade, Digital journal of CEPIS `http ://www.cepis.org/upgrade`
Editeur(s)	Josep Lladós-Canet, Jaime López-Krahe et Dominique Archambault *(Guest Editors)*
Volume/Date	Vol. VIII (2), Avril 2007
Pages	43–53

Cet article présente un état de l'art relativement exhaustif de ce qui se fait dans le domaine de l'accès aux jeux vidéo pour les personnes handicapées visuelles. Il a été publié dans Upgrade, la revue en ligne du CEPIS. Une monographie en espagnol a été publiée dans la revue Novática.

Voir mémoire section **2.3.4**, page 23

Computer Games and Visually Impaired People

Dominique Archambault, Roland Ossmann, Thomas Gaudy, and Klaus Miesenberger

The accessibility of computer games is a challenge. Indeed, making a computer game accessible is much more difficult than making a desktop application accessible. In this paper first we define game accessibility, then we present a number of papers published in the last decade: specific games (audio games, tactile games etc), games designed for all, and a few words about game accessibility and then we will describe the work that we are currently carry out in order to propose a framework to allow mainstream games accessibility.

Keywords: Accessibility, Computer Games, Visually Impaired People.

1 Introduction

Computer games have become an important part of child and youth culture, and most children, in developed countries, have a considerable experience of such games. Additionally these games are used by a growing part of the population, especially young adults (in average 25 years old, including 40% of women[1]) but the proportion of players is also growing in other age groups.

Indeed the mainstream commercial market for computer games and other multimedia products have shown a rather impressive development in recent years. For instance in 2002, costs for the development of a games could vary between 300,000 euros for a game on a wearable device, to 30 millions for the biggest productions (involving nearly a hundred of employees) [1] [2] [3]. Since 2002, the anticipation by players for more impressive games have caused budgets to increase, with increased use of new technologies.

People who cannot use the ordinary graphical interface, because they are totally blind or because they have a severe visual impairment (sight rated < 0.05), do not have access or have very restricted access to this important part of the youth culture [4]. This is particularly unfortunate for two main reasons. The first is that this group of people is probably the one who can benefit the most from technology. Indeed, technological tools benefits them in a lot of situations in their daily lives, at school as well as at work or at home, in mobility, etc. Therefore it seems important that children get used to using technology as early as possible. A second reason is that handicapped children can benefit a lot from the use of computer games for their psychomotor and cognitive development [5].

[1] TNS Sofres, *Le marché français des jeux vidéo* (The market of video games in France). afjv, November 2006. <http://www.afjv.com/press0611/061122_marche_jeux_ video_ france.htm>.

Authors

Dominique Archambault has a PhD in Computer Sciences and is an Associate Professor at the *Université Pierre et Marie Curie*, Paris. Since 1996 he has been working in the field of assistive systems for visually impaired people. He focuses on projects concerning non visual interfaces, Web accessibility, and educational tools for blind children. One of his main topics of interest is the way in which computer technologies can be used as a tool for children's development, particularly in the case of visually impaired children with additional difficulties (problems arising from their visual impairment or additional disabilities). He has coordinated 2 IST European projects (TIM - IST-2000-25298 and Vickie - IST-2001-32678). <dominique.archambault@upmc.fr>.

Roland Ossmann studied software engineering and is currently a PhD student at the Institute "integrated study" of the University of Linz. His research topic is computer games accessibility. Based on his development of a game designed for all, he is working on development guidelines and tools for computer game designers. He designed a web site on his guidelines for game accessibility. <roland.ossmann@students.jku.at>.

Thomas Gaudy studied at the National School of Video Game and Interactive Media (ENJMIN – France) and is now is preparing a PhD entitled "Video Games for visually impaired people". He has been developing four audio games which he is testing with users: two musical maze games, another one which is playable with a mouse, while the fourth one combines accessibility for visually impaired people and for people with motor impairment. <tomgody@yahoo.fr>.

Klaus Miesenberger is the director of the Institute "integrated study" of the University of Linz. After being guest professor at the *Université Claude Bernard, Lyon II*, he received his professorship in Human-Computer Interaction for People with Disabilities in 2001. He has been responsible for and worked on more than 25 national and international research, development and application projects in AT and ICT for people with disabilities. He has been responsible for the ICCHP (International Conference on Computers Helping People with Special Needs) since 1999. <klaus.miesenberger@jku.at>.

To give people with visual disabilities the chance to have access to multimedia games should be seen as an important issue for better inclusion and participation in society.

Designing games that work for visually impaired children is quite a challenge since the main feedback channel in games is usually visual. Indeed even if audio is more and more used in mainstream games, it has only a complementary role in a huge majority of cases. It improves the experience of the player but it is usually not bringing necessary pieces of information that the player would not get visually. For instance most of these games can be played efficiently with sound switched off.

This is probably the reason why very few computer games are accessible and very few have been developed for the visually impaired. Before 2000, the only accessible games that one could find were developed by the Swedish Library of Talking Books[2] .

2 Game Accessibility

Accessibility of games is a more complex problem than software accessibility in general. First, and it seems obvious, it is very important that accessible games still be games.

Accessible games must still be games!

2.1 Computer Access

Computer access for visually impaired people started to develop largely during the 80s, thanks to the appearance of personal computers. It was made possible by the development of speech synthesis and by the appearance of Braille refreshable displays. These tools allow to render information using alternative modalities (namely audio and touch).

To access desktop application, blind people have to use specific software applications, called screen readers, which literally reads the information on the screen to render it in an appropriate way using speech and/or Braille, according to the user settings. These applications were relatively simple when we were using console operating systems (text only on screens showing 25 lines of 80 characters), but became extremely complex with the development of graphical user interfaces. Nowadays screen readers are able to deal with multiple windows, to access client contents as well as menus, dialog boxes, etc. They allow a trained user to access efficiently most desktop textual applications (word processors, spreadsheets, Internet browsers, email software, etc), as well as system information and settings.

The improvement of screen technology during the 90s has allowed the development of enlargement software (screen magnifiers), which enlarge text or pictures up to 32 times, which is often a necessity for partially sighted people. Furthermore it is possible to change colours, resolution and also to redesign the organisation of information on the screen. Used alone, or in combination with speech synthesis, these software solutions enables people with really low vision to use the graphical display in an efficient way.

Additionally a number of specific devices which use the tactile modality or the haptic modality can be used together with specific software. First tactile boards are devices on which overlays can be inserted. The device transmits to the computer the location where it is pressed. Usually they are used to adapt computer games for young children. Indeed the overlays can support very rich tactile representations and therefore be very attractive for children. Tactile overlays may be prepared using various technologies, like thermoform, swallowed paper, polymerised glues etc, or simply by sticking different kinds of material. However, the major limitation with this kind of device is that is that overlays are static and not so easy to insert properly in the device.

Other tactile devices have appeared recently. These are tactile graphical refreshable displays which use the same kind of pins than Braille displays, to represent a graphic surface, with 16x16 pins or more[3] . These are still very expensive and for now mainly used in research labs, but they might come to the market in the next decade.

Other experimental devices called tactile transducers use the principle of deforming the skin by tangential traction [6]. An experimental device was designed comprising a 6x10 actuators array with a spatial resolution of 1.8x1.2 millimetre. It was successfully tested by subjects who were asked to detect virtual lines on a smooth surface.

Haptic devices include a variety of devices like vibrato tactile joysticks and gamepads, or the Senseable Phantom[4] . The Phantom is an input/output device which looks like a mechanical robot arm that is holding a stylus. When holding the stylus in the hand it is possible to touch, feel and manipulate virtual objects with 6 degrees of freedom. It uses only one point of interaction (the stylus), which means that the user will access the virtual environment through this single point, like touching objects with a stick [7].

Finally switches are very simple input devices which are used by people with very reduced motor capacities. They allow only to send one event (when pressed). They are very sensitive and allow a lot of precise settings.

2.2 Computer Accessibility

Unfortunately these specific access software applications (screen readers and screen magnifiers) are not able to access all software regardless of how it was developed. Indeed access software applications need to collect accessible information from the applications in order to render it using alternative modalities. This accessible information is mainly textual which means that all graphics that have a function must have alternative texts, but it is also necessary to get the information when something happens on the screen (when a dialog box appears, when some information arrives somewhere in the client window, etc...). If this information is not present in an application, the use of screen

[2] TPB, <http://www.tpb.se/english/computer_games>.
[3] See for instance Handytech GWP (*Graphic Window Professional*): <http://www.handytech.de/en/normal/products/for-blind/gwp/index.html>.
[4] Senseable Technologies Inc.

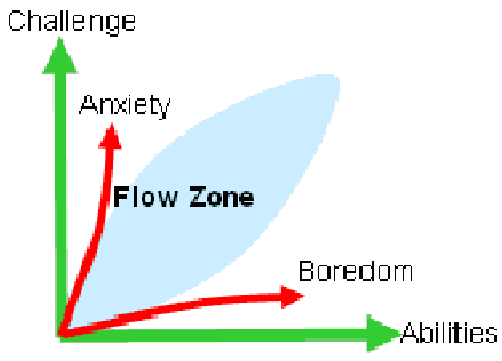


Figure 1: About Flow in Games.

readers with this application is at the best very difficult and in the most cases completely impossible.

To make applications accessible, accessibility frameworks have been developed and are available in the main environments. For instance, Microsoft has developed Microsoft Active Accessibility[5], which is a COM-based technology that provides assistive technology tools with access to the relevant information. On one side it is a collection of dynamic-link libraries that are incorporated into the operating system, on the other side is a COM interface and application programming elements.To make their applications accessible, application developers have to implement the IAccessible interface.

There exist similar frameworks on Mac[6] and on Linux desktop environments: Gnome accessibility project[7], KDE accessibility project[8].

Furthermore specific development frameworks need to support accessibility. For instance the Java Accessibility API[9] or Access Mozilla[10].

2.3 Accessibility of Contents

It is not enough that applications respect accessibility standards. In most cases the contents must be accessible too.

For instance in the case of a web site, the accessibility of web browser is necessary but the web contents must be accessible too. Graphical elements for instance must have textual alternatives, and this depends on the content itself. In that respect, the W3C launched the Web Accessibility Initiative to developed guidelines for Web Accessibility: WCAG[11]. (*Web Content Accessibility Guidelines*). These guidelines indicate how to use each of the HTML tags to make a web site accessible. For instance to support graphics, it is requested to insert alternative text within an "ALT" attribute on each IMG element.

2.4 Differences in the Case of Games

These accessibility solutions work satisfactorily for desktop applications but not for computer games. First the notion of "working satisfactorily" is a) not enough and b) not easy to define in that context.

Indeed, the results of a game can not be easily quantified unlike classical desktop applications. In word processing software, it is easy to measure the time needed by a user to write a document or to edit a document produced by a colleague. In a game we can observe if a player succeeds, and measure the time to finish a level or any case relevant for the game considered, but this is not enough. Unlike others software, games have to provides special feelings to players. There are likely to be some emotional factors to consider in the desktop applications, but they are usually not taken into account, or at least unless they affect the productivity. In the case of a game these factors are the most important.

It is not easy to describe the nature of these types of feelings: it is a large field of research and studies that describes it through two main concepts: the concept of presence and the state of flow. Both describe a state where the player feels totally immersed in the game universe. The presence concept is the ability of the game to provide the illusion for the player that he/she is in the virtual environment [8]. The presence sensation can be evaluated according to the goals of the game [9]. The efficiency of the interaction is also an essential point [10]. There are many others ways to approach this particular type of sensation felt by the player.

The other concept is the state of flow. This concept could appear very similar to the presence concept, but it is more oriented to a sensation of intense pleasure rather than an illusion of being elsewhere. It can be defined as a state of concentration, deep enjoyment and total absorption in an activity [11]. Flow is the result of the balance between two different psychological states, anxiety and boredom, themselves produced by the gathering of two aspects of gaming: the challenge of a task versus the abilities of the player [12] (see Figure 1).

In other terms, as we stated at the beginning of this section: Accessible games must still be games! Adults in work situations accept relatively large constraints on usability in order to be able to use the same software as their sighted colleagues and to work on the same documents.

The example of ord processing software provides a quite interesting way to illustrate this. Usually the simple access to character properties (like knowing if a word is in bold face or not) necessitates some complex manipulations. What is immediate by sight on any WYSIWYG word processing application, necessitates selecting the word, opening a dialog window and then navigating the various boxes of the character properties dialog in order to check the various settings currently selected. There exist other software solutions for word processing, like LaTeX, which allow much

[5] <http://msdn.microsoft.com/at>.
[6] <http://www.apple.com/accessibility/>.
[7] < http://developer.gnome.org/projects/gap/>.
[8] < http://accessibility.kde.org>.
[9] <http://www-03.ibm.com/able/guidelines/java/javajfc.html>.
[10] <http://www.mozilla.org/access/>
[11] <http://www.w3.org/TR/WAI-WEBCONTENT/>.

simpler access to this kind of information for a blind person, but they are usable only if the colleagues of this person use the same software! Otherwise, it's always more important to be able to use the same software as the others even if the usability is not so good.

This is not the case for children, especially when playing. In other words it is not enough to find a technical way of allowing to access to all information needed in the interface, the result must be as interesting and as usable as the original game.

Another important reason, which is as important as the previous one, is that it must be possible to succeed! Once again it seems obvious. In the example of a screen reader with a desktop application the user has to explore the content of the screen or of a window to know its contents. In the case of a game, let us consider a shooting game for instance, if an enemy enter the screen, a sighted person will perceive it visually within seconds, together with its location. But if a blind user needs to scan the screen to get the same information, the enemy will have plenty of time to defeat him/her and the game will be over very soon.

In the current mainstream market a huge majority of computer games are totally inaccessible to blind users as well as to most partially sighted users, and also to people having a large variety of other impairments.

2.5 Playable Alternative Modalities

To handle the special needs of the impaired players, new ways of interacting and communicating have to be found. The different modalities available for visually impaired users have to be studied in order to see what kind of interaction can be made with each device and how.

2.5.1 Audio

The first modality which comes to mind is obviously the audio. Considerable progress has been made in audio possibilities of computers, and what has largely contributed to this progress is the need for the development of computer games. These audio possibilities have been used for a quite large number of audio games, games that rely only on audio. We'll see a number of them in Section 3.1.

Unfortunately these developments have been exclusively driven by use cases where the audio is a complement to a visual experience, which is supposed to enhance the feeling of the situation for the player.

2.5.2 Tactile

The tactile modality can be used to design tactile or audio-tactile games. We'll see in Section 3.2 some games using tactile boards. Braille devices are usually not used to design games. Nevertheless some research is currently carried out in order to find models to represent a 2D space on a linear Braille display [13]. A few experimental games were designed in order to evaluate these models, for instance a snake game and a maze game.

2.5.3 Haptics

Sjöström [14] studied the possibilities offered by haptic technologies for creating new interactions usable by blind people. He worked especially with the Senseable Phantom.

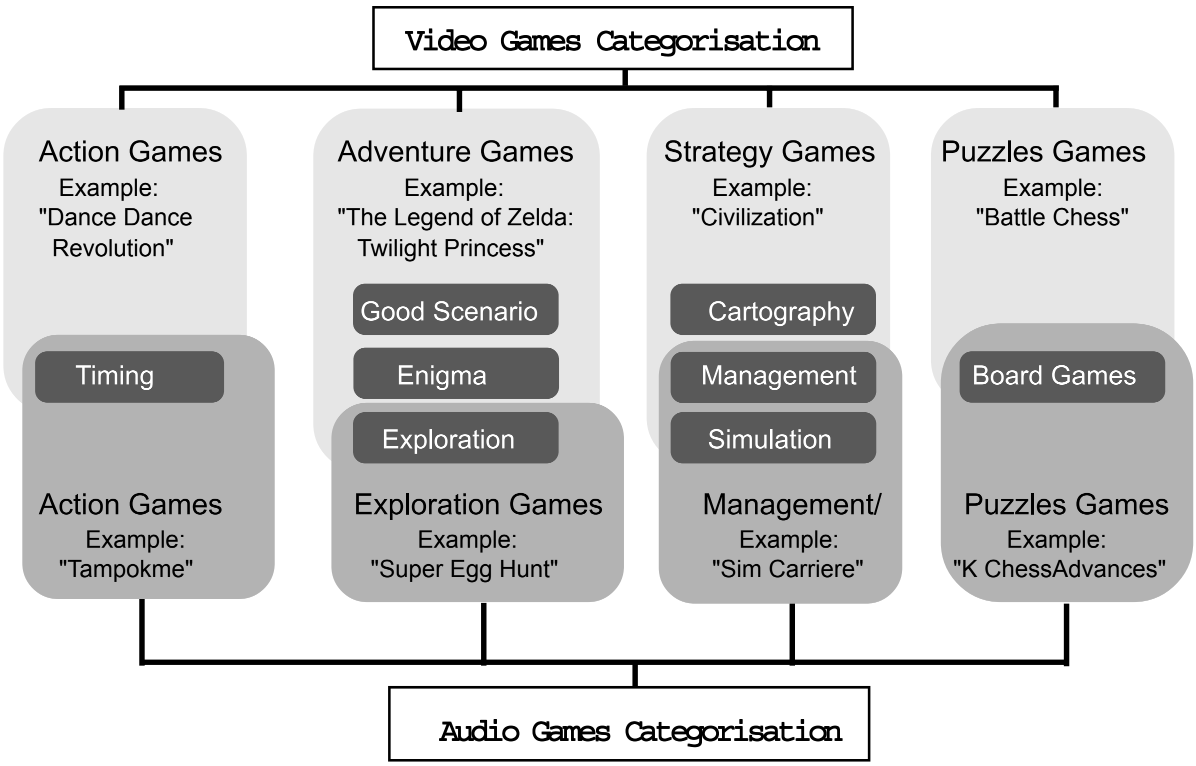


Figure 2: Comparison of Video Games Categorisation and Audio Games Categorisation.

Johansson and Linde [15] used an inexpensive force feedback joystick in a virtual maze. Wang et al. [16] have designed an experimental memory game in order to assess the possibilities of their tactile transducers. Raisamo et al. [17] present the same kind of memory game using a low cost vibrato-tactile device (a force feedback gamepad). In these three cases, the interest of the researchers is not in the game development itself, but the use of the game as a tool to study the haptic perception. Nevertheless the players were able to successfully play the games and showed an interest in the devices, so use in future accessible games would be possible.

Evreinov et al. [18] designed a vibro-tactile pen and software to create tactons and semantic sequences of vibro-tactile patterns on mobile devices (iPAQ pocket PC). They propose special games to facilitate learning and manipulation by these tactons. The techniques are based on gesture recognition and spatial-temporal mapping for imaging vibro-tactile signals.

The Phantom was used in several game experiments. Aamisepp and Nilsson [7] built a plugin for an open source 3D game engine (Crystal Space) that allows navigation in the 3D environment and to touch the surroundings with the Phantom. The authors conclude that their system can be used with 3D environments specially built for the haptic environment but that it would not really work with existing environment used in 3D games like Half-life or Quake. The reason is that the Phantom uses a single point of interaction, which makes it very difficult to understand a 3D shape.

Crossan and Brewster [19] describe research about the possibility of using two hands with two different devices. They were using their dominant hand to move a cursor with a phantom and their other hand to receive tactile information from a device offering a 4x8 array of pins[12]. A maze game was designed for the experimentation. They conclude that most users could solve the task with little training, which tends to show that such combination can be used to make accessible games or accessible versions of games.

The Phase project [20] is an experimental game using haptic feedback. It has been successfully played by visually impaired people in various exhibitions. The game refers to function of a phonograph: Phase offers visual stereoscopic landscape wherein a player moves a kind of play head that produces sounds, according to its position over the landscape. It can be used as a musical toy that users just listen to or as a challenging game. To beat the game, players have to increase speed by trying to reach the lowest location of the landscape. The haptic device (Senseable Phantom) allows the user to feel the relief and to find acceleration zones. It enables visually impaired players to successfully handle the game.

[12] Virtouch VTPlayer.

[13] <http://www.guitarherogame.com>.

[14] <http://www.terraformers.nu>.

About the Phantom, we must also mention BattlePong, which is a pong game where the players use a Phantom to handle the racket. BattlePong was presented at the Experimental Gameplay Workshop at GDC 2004 (*Game Developers Conference*).

3 Specific Games

This section focuses on games specifically designed for visually impaired people. These games are usually funded by foundations or non-profit organisations. Most of them are very nice for visually impaired people but have little interest for the mainstream, except maybe a few of the audio games.

3.1 Audio Games

Audio games are games, in which the main play modality is audio, which includes three different concepts. The first involves mainstream video rhythm games like "guitar hero II"[13]. The second is related to artistic musical experimentations. The web site audiogame.net refers interesting interactive works. In this paper we will focus on the third concept - audio games which can be played with audio only and without visuals which are therefore accessible to visually impaired players.

In 10 years, over 400 audio games have been developed, which is a very small number as compared to video games. Indeed, compared to the market of computer games, audio games are a tiny market, profitability of such projects is smaller and development teams contain generally only between one and four persons. Nonetheless, the production rate has dramatically increased those last few years and the community of players and developers strengthens with sites such as Audio games.net, which seems to imply that this new medium is in a very promising phase of expansion.

Researchers could have an important role to play in that expansion process, by contributing to develop innovative features like a more pleasant audio rendering, by projecting audio games in a futuristic point of view by using uncommon technology like GPS for instance and by participate in the elaboration of games which can interest both the sighted and visually impaired community [21]. There exist a few visual audio games which can be very impressive and playable as well with or without sight. "Terraformers"[14] [22] was developed with accessibility as part of the original concept of the game. On the other hand, "Audioquake" [23] (see Section 5.5) was developed as a research project to make a non accessible game accessible.

3.1.1 Audio Games Categorisation

Audio games, like video games, can be categorised. The categorisation of audio games provides conceptual tools allowing to comprehend better the interaction mechanisms used for the players to have fun. In order to categorise audio games, we use a categorisation of video games as a theoretical basis of thinking [3]. Natkin distinguishes four main types of games that can be played alone: action games, adventure games, puzzle games and strategy games. These categories will be described more precisely below. Natkin mentions that these types of games can be mixed together and it is true that numerous video games belong to several categories.

The categorisation of audio games proposed in [24] looks similar, with four matching categories, but they are not exactly the same as video games categories (see Figure 2). Indeed audio games do not have the same history as video games. They have not had the same evolution. Audio games owe their existence to video games but also to textual games. Without the visual, texts are the best means to give the players the rules of the game. Audio games with more complex rules, such as parlour games or management/ simulation games, are therefore naturally more textual than audio. Games with more simple rules, such as action and exploration games, have more recourse to non-verbal sounds. Most of the games that are more non-verbal audio than textual can be considered as a mixture at different levels of the interaction mechanisms of action and exploration games. Some interactions have to be based on a precise timing (temporal dimension of the interaction) and some interactions permit the exploration of a geographic space. Thus, current audio games can be studied with three factors: their dependency on verbal information, their dependency on interaction mechanisms based on timing and their dependency on the interaction mechanisms of exploration.

3.1.2 Action Games

Action games directly refer to games coming from and inspired by arcade games and requiring good dexterity from the player. One example of such games is "Dance Dance Revolution" (DDR). DDR is the first of an important series of musical video games. Playing these games requires a good synchronisation with a tempo given by the music. They are not accessible because the kind of action to be done is given visually. The visual information allows the player to anticipate the time of the interaction. The tempo of the game can be rather fast and really challenging. The concept of timing is very important but it depends at the same time on audio and visual perception.

There are also audio action games, where the success of the player depends on an interaction based on a precise timing. "Tampokme"[15] is such an accessible audiogame. The principles of the game are very simple compared to DDR. There is no possibility of anticipating but timing is essential. The player has to identify various kinds of audio signals and react accordingly and in fixed time. The state of flow is not as intense as in DDR but the potential for improvement is high and it is fully accessible to visually impaired people as well as to people who have mobility troubles (it can be played with a single button).

3.1.3 Adventure Games/Exploration Audio Games

Adventure games allow players to take part in a quest. This type of game combines three essential features: an interesting scenario, the exploration of new worlds and riddle solving. A typical example is the series "Zelda". The enigmas and the scenarios of this kind of game are important but not as much as the action of moving around of the avatar of the player in a virtual environment.

There are also adventure audio games, though the scenarios of these games, as compared to video games, are still very little developed and that the activity of puzzle solving is clearly secondary. The primordial aspect of these games is, most often, the exploration of a new geographic environment. That is why we call this category "exploration games" in the case of audio games.

"Super Egg Hunt"[16] focuses on the move of the avatar on a grid, where the players must locate objects only from audio feedback. A quite clever use of stereo, volume and pitch of the sounds allows an easy and pleasant handling.

3.1.4 Strategy Games/Management-simulation Audio Games

Strategy games require that the players control an army, a territory and resources by manipulating maps. The game "Civilization" allows the player to control a large number of characters, spread across a large number of places. The player manipulates a lot of parameters in order to ensure that the population survives and develops.

There are a few strategy audio games but the manipulation of maps is rather difficult without the visual aspect. It is still possible: galaxy ranger and AMA Tank Commander are two audio games that offers an interesting approach of strategy games, but they are isolated [25]. The aspects that are the most prominent for these games are the management of resources and simulation.

The game "simcarriere"[17] ignores the map aspect and focusses on the simulation/management side. The player has to manage a lawyer's office by buying consumables, hiring personnel, and choosing the kind of cases to defend.

3.1.5 Puzzle Games

Finally puzzle games are inspired from traditional games. There are puzzle audio games which directly refer to parlour games that are not accessible in their usual material form. Nevertheless audio and video puzzle games are quite similar in their principles.

The game "Battle Chess" is an adaptation of the Chess game into a video game. "K-Chess advance"[18] is also an adaptation of the Chess game, but focussing on audio. In those cases, the feeling of flow does not rely on principles of reactivity in front of audio or video signals, but on entertaining principles that have stood the test of time before the advent of computers.

[15] Tampokme, the audio multi-players one-key mosquito eater, CECIAA/CNAM CEDRIC/UPMC INOVA, <http://www.ceciaa.com/?cat=lois&page=action>.
[16] Egg Hunt, LWorks, <http://www.l-works.net/egghunt. php>.
[17] <http://www.simcarriere.com/>.
[18] KChess Advance, ARK_Angles, <http://www.arkangles.com/kchess/advance.html>.

3.2 Tactile Games

Tactile games have inputs and/or outputs that are carried out with tactile boards or Braille displays, usually in combination with audio feedback.

As mentioned in Section 2.5.2, the use of Braille displays for gaming is only experimental at the moment so we will only discuss projects that involve tactile boards.

Back in 1999, we developed a first software workshop[26] allowing to design software games using a tactile board for the input. A scripting language was used to define regions on the tactile board and to attach sounds to these regions. It was also possible to design several scenes and to link them together, which allowed the design of audio/tactile discovery games as well as matching games, quizzes and interactive stories.

Then during the TiM project [27] we developed several tactile games. One was an accessible version of "Reader rabbit's Toddler", where the children can feel tactile buttons on the tactile board, and then drive the game from this board. This adaptation has been designed with the support of the publisher of the original game.

This work allowed us to realise how important it is to compensate the lack of visual information in order to ensure that the adapted game has enough attractive feedback for the visually impaired children[28]. Indeed in a first version we had only adapted the tactile interfaces, so the game was usable by the children, but all the audio outputs were the original ones. It was clearly not enough, the children did not consider it as a game. In a second version we added a lot of audio content, which had been written by a professional author. This version was much appreciated by the children.

Another game developed within the TiM project was FindIt, which was a very simple audio/tactile discovery and matching game. It is intended for very young children or children with additional disabilities. The player must associate sounds with pictures on the screen or tactile information on a tactile board. Recorded comments and clues are associated with the items. From this game we developed a game generator [29] which is currently being evaluated by practitioners working with visually impaired children. The generator allows educators and teachers who work with visually impaired children to design their own scenarios and to associate them with tactile sheets that they design themselves manually.

Tomteboda resource centre in Sweden has published a report in which they try to stimulate parents and educators to develop their own games using tactile boards [30].

4 Games Designed for All

The goal of games being developed under the title of "designed for all" is to give players with a wide variety of abilities or disabilities the opportunity to play these games. This requires a very advanced game setting and configuration.

4.1 UA-Chess

UA-Chess[19] (*Universally Accessibility Chess*) [31] is a universally accessible Internet-based chess game that can be concurrently played by two gamers with different (dis)abilities, using a variety of alternative input/output modalities and techniques in any combination. It was developed by the Human-Computer Interaction Laboratory of ICS-FORTH in close cooperation with the Centre for Universal access and Assistive Technologies.

The game supports a collection of possible input devices like keyboard (including switch devices), mouse and even speech recognition (recognise more then one hundred words). With these input devices, several configuration and combinations are possible. For example the keyboard can be used, beside the standard input, for input scanning. Furthermore, keyboard shortcuts help the gamers to control the game and the game menus quickly.

The two possible output channels are visual and auditory through speech synthesis. An integrated screen reader makes it possible to play the game without visual information. The player can decide which information he/she wants to be read. UA-Chess gives the opportunity to save these customised information in profiles.

The game provides a default, a blind and two switch-input profiles. Furthermore, two user profiles will be supported. Certainly, each of the two players can select their own profile. As already mentioned, it is a two player game, which can be played together on one computer or over the Internet.

The programming language and technique is Macromedia Flash MX Professional 7. Flash was chosen because the plug-in is available for all important web browsers on several operating systems (Windows, Mac OS, Linux and Solaris). Flash does not support speech input or output, so to support speech recognition and synthesis *Speech Application Language Tags* (SALT) technology[20] was used.

4.2 Access Invaders

Access Invaders[21] is a designed-for-all implementation of the famous computer game Space Invaders, with the target groups of people with hand-motor impairments, blind people, people with deteriorated vision, people with mild memory/cognitive impairments and novice players [Grammenos et al., 2006]. Furthermore people belonging to more than one of the previous groups. Like UA-Chess, this game is developed by the Human-Computer Interaction Laboratory of ICS-FORTH. Universal Access will be achieved by supporting alternative input/output modalities and interaction techniques that can co-exist and cooperate in its user interface, in combination with configurable player profiles. Each game parameter can be adapted both based on the player's profile and the current game level. Non-visual gameplay is also supported by full acoustic rendering of game information and a built-in screen reader.

[19] <http://www.ics.forth.gr/hci/ua-games/ua-chess/>.
[20] <http://www.saltforum.org>.
[21] <http://www.ics.forth.gr/hci/ua-games/access-invaders>.

Multi-player games are available, where people with different (dis)abilities can play cooperatively, sharing the same computer. In this case, the game's interaction parameters can be independently adjusted for each player. An unlimited number of concurrent players is supported. This is reached by the concept of the so called *Parallel Game Universe*. Different kinds of aliens (varied by amount and strength) belong to the different players. A player just can kill aliens belonging to him/her and vice versa.

Future development will cover the support of tactile output through a Braille display and a force feedback joystick and stylus. Furthermore, an interactive editor for user profiles and dynamic gameplay adaptation, which includes monitoring the player's actions and dynamically adjusting the gameplay to better match the player's skills.

4.3 Tic Tac Toe

The approach of Ossmann et al. [33] was not to develop a new game designed for all players. It was decided to make an already published game accessible and show, if this is possible and how big the effort is. An open source implementation of Tic Tac Toe was chosen. The accessible functionality bases on *Guidelines for the Development of Accessible Computer Games* (see Section 5.2) and was a good test for the usability and completeness of these guidelines. The so called *Descriptive Language* was used to realise the accessible features during the implementation.

This language connects alternative input/output devices with the game engine with the possibility to use several input devices simultaneously and present the game output on several output devices. This means that, as an example, the game player can decide if he/she wants to have all graphical objects presented on screen, described over speaker or be on both output devices. Additionally the Descriptive Language covers the option for an extensive configuration of the game, so that the game can be customised to the special needs of a (dis)abled person. This configuration ranges from game speed to the number of opponents to many other game options, depending on the kind of game. The Descriptive Language is, as the Tic Tac Toe game itself, still under development and this game is the first example, using this language.

The game provides, beside the already mentioned full sound output of all objects on the game field, also the possibility to use input scanning in connection with (one) switch devices. The future development will cover the support of visual impairments and full accessible game configuration.

5 Game Accessibility

The goal of games accessibility is to bring the idea of accessible games (or games designed for all) to the mainstream and show different approaches. Here is an overview of papers about research work and development on this topic.

5.1 Accessibility in Games: Motivations and Approaches

[34] is a white paper published by the Games Accessibility Special Interest Group[22] (GA-SIG) of the IGDA (*International Game Developer Association*) in 2004. Firstly they give the following definition of games accessibility: "Game Accessibility can be defined as the ability to play a game even when functioning under limiting conditions. Limiting conditions can be functional limitations, or disabilities such as blindness, deafness, or mobility limitations." Furthermore definitions of the different kinds of (dis)abilities are given followed by statistics about (dis)abilities in the population and the possibilities of game based learning.

The paper also covers a collection of hints, suggestions and guidelines for the development of accessible games. Moreover there is a collection of accessible games with a short description of each game and a categorisation of the (dis)abilities supported. A listing of assistive technologies and an overview of state of the art research completes the paper.

It is the first famous publication that woke up the mainstream game developers and showed them that it is possible and necessary to include more users in the mainstream games.

5.2 Guidelines for the Development of Accessible Computer Games

Guidelines are an important tools to bring games accessibility to the mainstream. Ossmann and Miesenberger [35] show a set of guidelines[23], based on the already mentioned guidelines from IGDA and a set of guidelines from the Norwegian company MediaLT[24], and their development process. The guidelines are a collection of rules, hints and suggestions as to how to develop accessible computer games, divided into the categories of level/progression, input, graphics, sound, and installation and settings.

The guidelines have, beside the rules themself, a categorisation in three classes of priorities: a) must have, which means, that it is absolutely necessary for the listed group of gamers. Otherwise the game is not accessible for them, b) should have, which means, that it is a big help for the listed group of gamers and c) may have, which means, that it is a helpful feature for the listed group of gamers. Furthermore there are four groups of (dis)abilities: visual, auditory, mobility and cognitive (dis)abilities. These (dis)abilities are allocated to the priorities, e.g. one rule can have priority 1 for visually impaired people and priority 3 for auditory impaired people. The next steps will be adding code samples and best practice methods to fulfil the rules. Another step will be making a stronger integration of assistive technologies in the guidelines, specially adding an input device section to them.

The future goal is to have a useful and usable set of guidelines for game development like the web accessibility guidelines for web pages.

[22] <www.igda.org/accessibility>.
[23] <http://gameaccess.medialt.no/guide.php>.
[24] <http://www.medialt.no>.

5.3 Computer Games that Work for Visually Impaired Children

Archambault et al. [28] describe the work on computer games for blind or severely visually impaired children from 3 to 10 years old. The research and development work was done during the TiM project, which was a co-operation of several European institutes and organisations, funded by the European Commission. Various mainstream games had been studied and several of them were adapted to be playable for the target group. Also new ones were developed. Several case studies were accomplished, one with the game "Reader Rabbit's Toddler". In the original version of this game, the player can have access to 9 different educational activities. From these 9 different activities in the original version, 4 could be adapted plus the navigation in the main menu.

All output was converted to alternative modalities (basically audio in this game) and for the input, a device called tactile board was used. This is an input device on which a rectangular sensitive area is divided into 256 cells. On top of the sensitive area, a tactile overlay may be inserted. Rich tactile overlays were designed, using pieces of various materials, stuck on a robust PVC sheet, and Braille labels.

A second game, called Mudsplat, which based on the classic arcade game "Space Invaders", was developed. The game has a various degrees of difficulty and traditional features of arcade games were implemented (like high score, extra lives, levels, bonus objects, bonus levels,...).

Furthermore the paper includes 14 rules for games, being developed for the already mentioned target group. These rules based on the results of research and test cases with children, including information about the game play itself, about the navigation in the menus and about content for the sighted.

5.4 Internet and Accessible Entertainment

[36] is based on a project, which is aimed at adapting and developing entertainment software for young people with (dis)abilities, including those with mental (dis)abilities, or a combination of severe motor handicaps and perceptual losses. Entertainment was chosen, because entertaining software may provide good motivation e.g. for learning assistive devices and standard computer skills. During the project, a new product should be developed following the principles entertaining, simple (not childish), accessible and flexible.

Internet games using Macromedia Flash were used based on the following considerations: support for different platforms, no client installation and only one source to maintain in combination with Flash's built-in tools for creating graphics and animations, use of mp3-files for music, the accessibility support is fairly good and the easy distribution to different medias like Internet and CD-ROM. The developed game (called "HeiPipLerke") can in short be described as a music composer adjusted to meet the requirements of the target group. It provides input scanning and audio output by default.

Some problems and challenges occurred, e.g. the implementation of input scanning and speech output, which was not available in Flash. Furthermore pointing to, and selecting items on the screen using a touch screen, followed by some problems changing the colours of all objects for people with visual impairments. The game based on the Guidelines for Accessible Games (see Section 5.2) and "Best Practices for Accessible Flash Design" guidelines. The future work on the game will include more content and support for the hearing impaired by including sign language in addition to speech.

5.5 Making the Mainstream Accessible: What's in a Game?

Atkinson et al. [23] describe the work of the AGRIP project - an effort to develop techniques for making mainstream games accessible to blind and visually-impaired gamers. *AudioQuake* is the first adaption of an existing mainstream game (Quake from id-Software) designed specifically for sighted people that has been made playable for blind gamers by adding an "accessibility layer". During the development of this layer, one of the issues to be solved was navigation. Navigation issues were divided in global navigation towards one's ultimate goal (e.g. the red team's flag, ...) and local navigation (e.g. how do I get out of this room?). At the low-level game accessibility stage of the project's development, the primary concern was to develop the support of effective local navigation.

One "tool" that was implemented for the local navigation was the EtherScan Radar. It warns players of nearby enemies and team mates using a RADAR-like metaphor: sounds emanate from the position of the enemy and have a gain and repetition speed proportional to the players distance from them. Another topic is serialisation. It reduces multidimensional problems into single-dimensional problems, which is important for the two most popular accessible output formats — speech and 1-dimensional Braille displays. Serialisation has a close connection to prioritisation. This means, that different kinds of information (enemies, walls, ...) will get different kinds of priorities when being serialised. Several priority paradigms are discussed there.

6 Game Accessibility today

We have seen (Section 3) that a number of games have been developed specifically for visually impaired users. These are mostly audio games but a few tactile games exist. These games have two points of interest: firstly of course they are of tremendous interest for visually impaired players (especially since the number of such games is extremely limited regarding the number of mainstream games), and secondly in the research field there are many experiments or demonstrations of how to render various interaction situations with alternative modalities. This can be completed by a number of research papers about new uses of modalities in the game play (Braille devices, haptic...) described in Section 2.5.

In the specific case of audio games some more progress could be achieved. Indeed they still don't use the specific musicality aspects of sounds. Multi-player audio games are required for better communication between players [25]. In order to progress it is necessary that audio games start to interest a larger community of players.

Then in Section 4 we have presented 3 research projects focussed on games designed for all. These games must be seen as good practice examples, demonstrating that Universal Access is a challenge and not utopia. In these projects we have to admit that the alternative access to these games requires more development than the rest of the game.

We assist the way to a general awakening to accessibility needs by the mainstream game developers. Some first efforts were started to improve accessibility of mainstream games, with particular guidelines supported by the International Games Developer Association. If they are ever partially implemented we can expect that the general accessibility level will be raised, which means that for a large number of people having a slight to moderate visual impairment, more games will be playable.

If we want to go further in this field, which means we want to make more mainstream games accessible, and if we want to make them accessible to all, we need support for accessibility included in the mainstream games themselves. This is the only way that will allow the building of accessible interfaces for mainstream games.

For now, considering Windows desktop applications, we have seen in Section 2.2 that it is necessary to have accessibility support embedded in the applications in order to make them accessible to alternative interfaces, supporting alternative modalities, like screen readers. This was achieved through the Microsoft Active Accessibility, but if this technology is well suited to desktop text-based applications it is clearly not enough for games (see Section 2.4).

In the close future we need to design a framework that will allow development of accessible solutions for games, that we call *Active Game Accessibility* (AGA). The goal of AGA is to allow mainstream game developers to offer support to accessibility solutions without significantly increasing the development charge for those games. This support will be used by Accessibility specialists to design accessibility solutions. These solutions will be various, depending on each game.

For instance AGA will allow the design of some kind of game screen readers which will make the most simple games accessible. The AGA framework will concern a much larger group than visually impaired people with support for all kind of disabilities. Then specific access software for various types of disability can be developed.

These generic accessibility software solutions will be necessary but will not be enough to make any kind of games accessible. Indeed in the most complex cases it will still be necessary to develop specific interfaces to the games. We have seen in Section 3.2 that in most cases it is necessary to at least add some audio feedback in order to compensate the lack of visual information. In more complex games it will be necessary to adapt the game-play itself. The AGA framework will allow the design of such interfaces that can communicate with the mainstream games themselves.

Today more and more actors of the game industry are aware that something must be done in the near future to improve the accessibility of computer games. We can be reasonably optimistic since we set up collaborations with specialists of Game Accessibility (for all disabilities), with several companies working in the mainstream, and with Assistive Technology providers. The game is not over yet!

References

[1] S. Natkin, C. L. Prado, V. Gal, L. Vega. *Quelques aspects de l'économie du jeu vidéo*. In *Actes des journées d'étude: Internet, jeu et socialisation*, Paris, France. Ossmann et al., 2006.

[2] F. Fries. *Propositions pour développer l'industrie du jeu vidéo en france*. Rapport à l'attention de M. Francis Mern Ministre de l'Économie, des Finances et de l'Industrie, et de Mme Nicole Fontaine, Ministre déléguée à l'Industrie, 2003.

[3] S. Natkin. Video games and Interactive Media: A Glimpse at New Digital Entertainment. AK Peters.

[4] A. Buaud, H. Svensson, D. Archambault, D. Burger. Multimedia games for visually impaired children. In K. Miesenberger, J. Klaus, W. Zagler, editors, Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs), Vol. 2398 of LNCS, pp 173 - 180, Linz, Austria. Springer.

[5] A. Hildén, H. Svensson. Can All Young Disabled Children Play at the Computer? In K. Miesenberger, J. Klaus, W. Zagler, editors, Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs), Vol. 2398 of LNCS, Linz, Austria. Springer, 2002.

[6] Q. Wang, V. Hayward. Compact, Portable, Modular, High-performance, Distributed Tactile Transducer Device Based on Lateral Skin Deformation. In Symposium on Haptic Interfaces For Virtual Environment And Teleoperator Systems, pp 67-72, Arlington, VA. IEEE VR, 2006.

[7] H. Aamisepp, D. Nilsson. Haptic hardware support in a 3d game engine. Master's thesis, Department of Computing Science Lund Institute of Technology, 2003.

[8] B. G. Witmer, M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. Presence, 7(3):225-240, 1998.

[9] A. Gaggioli, M. Bassi, A. D. Fave. Quality of experience in virtual environments. In G. Riva, F. Davide, W. IJsselsteijn, editors, Being There: Concepts, Effects and Measurments of User Presence in Synthetic Environments, chapter 8, pages 121-136. Ios Press, Amsterdam, The Netherlands, 2003.

[10] X. Retaux. *Presence in the environment: theories, methodologies and applications to video games.* PsychNology Journal, 2003.

[11] D. Johnson, J. Wiles. Effective affective user interface design in games. Ergonomics, 46(14):1332-1345, 2003.

[12] C. Jenova. Flow in games. MFA Thesis, University of Southern California. <http://www.jenovachen.com/flowingames/>, 2001.

[13] A. Sepchat, N. Monmarché, M. Slimane, D. Archambault. Semi automatic generator of tactile video games for visually impaired children. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp 372-379, Linz, Austria. Springer.

[14] Sjöström. The sense of touch provides new computer interaction techniques for disabled people. Technology and Disability, 1999.

[15] A. Johansson, J. Linde. Using Simple Force Feedback Mechanisms as Haptic Visualization Tools. In Proc. of the 16th IEEE Instrumentation and Measurement Technology Conference, 1999.

[16] Q. Wang, V. Levesque, J. Pasquero, V. Hayward. A Haptic Memory Game using the STRESS2 Tactile Display. In CHI2006, Montréal, Québec, Canada. ACM.

[17] R. Raisamo, S. Patomäki, M. Hasu, V. Pasto. Design and evaluation of a tactile memory game for visually impaired children. Interacting with Computers, 19(2):196-205, 2007.

[18] G. Evreinov, T. Evreinova, R. Raisamo. Mobile games for training tactile perception. In ICEC 2004 - Third International Conference on Entertainment Computing, volume 3166 of LNCS, pages 468-475, Eindhoven, The Netherlands. Springer.

[19] A. Crossan, S. Brewster. Two-handed navigation in a haptic virtual environment. In Extended Abstracts of CHI2006, Montréal, Québec, Canada. ACM.

[20] X. Rodet, J.-P. Lambert, R. Cahen, T. Gaudy, F. Guedy, P. Mobuchon. Study of haptic and visual interaction for sound and music control in the Phase project. In Proceedings of the 2005 conference on New interfaces for musical expression, pages 109-114, Vancourer.

[21] E. Velleman, R. van Tol, S. Huiberts, H. Verwey. 3D Shooting Games, Multimodal Games, Sound Games and More Working Examples of the Future of Games for the Blind. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Vol. 3118 of LNCS, pp 257-263, Paris. Springer.

[22] T. Westin. Game accessibility case study: Terraformers – a real-time 3d graphic game. In Proc. of the The Fifth International Conference on Disability, Virtual Reality and Associated Technologies, Oxford, UK, 2004.

[23] M. T. Atkinson, S. Gucukoglu, C. H. C. Machin, A. E. Lawrence. Making the mainstream accessible: What's in a game? In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pages 380-387, Linz, Austria. Springer.

[24] T. Gaudy, S. Natkin, D. Archambault. *Classification des jeux sonores selon leur type de jouabilité*. In Proceedings of Handicap 2006 Conference, pp 221-226, Paris.

[25] M. Feir. A Decade's Accessible Gaming: An Old Editor's Anniversary Observations, 2006. Audyssey, 49.

[26] D. Archambault, D. Burger. Development Tools for Non Visual Education Applications. In Proc. CSUN'99 (14th Annual Conference "Technology and Persons with Disabilities"), Los Angeles, California, USA, 1999.

[27] D. Archambault. The TiM Project: Overview of Results. In K. Miesenberger, J. Klaus, W. Zagler, D. Burger, editors, Proc. ICCHP 2004 (9th. International Conference on Computers Helping People with Special Needs), Lectures Notes in Computer Science 3118, pp 248 - 256, Berlin. Springer.

[28] D. Archambault, D. Olivier, H. Svensson. Computer games that work for visually impaired children. In C. Stephanidis, editor, Proceedings of HCI International 2005 Conference, Las Vegas, Nevada. 8 pages (proceedings on CD-Rom).

[29] A. Puret, D. Archambault, N. Monmarché, M. Slimane. A simple game generator for creating audio/tactile games. Technology and Disability, 18(4):227-236, 2006.

[30] J. Hammarlund. Computer play for children who are severelly visually impaired: Using an alternative keyboard with tactile overlays. Technical Report 20, Tomteboda resource centre, Stockholm, Sweden, 1999.

[31] D. Grammenos, A. Savidis, C. Stephanidis. Uachess: A universally accessible board game. In Salvendy, G., editor, Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction, Las Vegas, Nevada, 2005.

[32] D. Grammenos, A. Savidis, Y. Georgalis, C. Stephanidis. Access invaders: Developing a universally accessible action game. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pages 388-395, Linz, Austria. Springer.

[33] R. Ossmann, D. Archambault, K. Miesenberger. Computer game accessibility: From specific games to accessible games. In Proceedings of CGAMES'06 Conference (9th International Conference on Computer Games), Dublin, Ireland, 2006.

[34] IGDA, I. G. D. A. Accessibility in games: Motivations and approaches. <http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf>, 2004.

[35] R. Ossmann, K. Miesenberger. Guidelines for the development of accessible computer games. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp 403-406, Linz, Austria. Springer.

[36] M. Tollefsen, A. Flyen. Internet and accessible entertainment. In K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer, editors, Proc. ICCHP 2006 (10th. International Conference on Computers Helping People with Special Needs), Vol. 4061 of LNCS, pp 396-402, Linz, Austria. Springer.

Chapitre 2

MaWEn

Les trois articles de cette section concernent les travaux sur la compréhension d'expressions mathématiques et l'aide aux calculs mathématiques. MaWEn – *"Mathematical Work Environment"*– est le nom d'un ensemble de prototypes dont la plupart ont été réalisés et évalués dans le cadre du projet Micole. Le premier article présente les stratégies que nous avons utilisé pour faciliter la compréhension d'expressions mathématiques et l'accès simultané à plusieurs interfaces avec pointage transmodal. Le second présente des prototypes d'assistants permettant d'aider les élèves dans les calculs, en tentant de suppléer aux graffitis utilisés par les voyants sur la feuille de papier. Enfin l'article présenté en section 2.3 décrit le modèle que nous avons mis au point pour supporter ces fonctions d'assistance.

2.1 [WALT/ICALT'2007]

Titre	**Mathematical Working Environments for the Blind : What is needed now ?**
Auteur(s)	Dominique Archambault, Bernhard Stöger[(a)], Mario Batušić[(a)], Claudia Fahrengruber[(a)] et Klaus Miesenberger[(a)]
Affiliations	[(a)]Johannes Kepler Universität Linz, Autriche
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of ICALT Workshop on Advanced Learning Technologies for Disabled and Non-Disabled People – CEUR Workshop Proceedings 357
Editeur(s)	James Ohene-Djan et Marion Hersh
Lieu/Date	Niigata, Japon, Juillet 2007
Pages	7–16

Une version courte (2 pages) a été incluse dans les actes de la conférence ICALT (*7th IEEE International Conference on Advanced Learning Technics*) auquel ce workshop était joint.

Voir mémoire section **3.1.2.1**, page 27

Workshop on Advanced Learning Technologies for Disabled and Non-Disabled People (WALTD)
at
The 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007) July 18-20, 2007, Niigata, Japan

http://www.icaltd.doc.gold.ac.uk/

Mathematical working environments for the Blind: what is needed now?

Dominique Archambault[(1)], Bernhard Stöger[(2)], Mario Batušic[(2)], Claudia Fahrengruber[(2)] and Klaus Miesenberger[(2)]

[(1)] INOVA/UFR922,
Université Pierre et Marie Curie, Paris, France
dominique.archambault@upmc.fr

[(2)] Integriert Studieren,
Johannes Kepler Univsersität, Linz, Austria
Bernhard.Stoeger@jku.at

Abstract

Blind people encounter great difficulties in dealing with Mathematics. Based on an analysis of these problems, we shall outline possible strategies to overcome them through software support. These strategies are currently being implemented in prototypes developed for a new Mathematical Working Environment dedicated to blind pupils and students.

1. Introduction

The study of Mathematics has always been particularly difficult for blind individuals. Indeed we can observe that a large majority of blind pupils do not succeed in Maths studies, while the average mainstream pupil succeeds more easily. As Maths is crucial in most science disciplines, this limits study options and future job opportunities for blind people.

We assert that there is no reason that Mathematical semantics can not be understood because of blindness; rather the biggest barrier is access to Mathematical content, which can only be through speech or Braille.

Most Mathematical concepts are best explained using drawings and notes which illustrate the main content (1).

$$(x+1)(x-1)=x^2-\cancel{x}+\cancel{x}-1=x^2-1 \quad (1)$$

In addition, the Mathematical notation itself uses 2 dimensions in order to convey the general structure of the formula rapidly, making the semantics easier to understand. One "pictures" the basic Mathematical content at a glance, which helps reading the details more efficiently, since the role of every part of the expression is already assimilated.

$$\frac{x+1}{x-1} \quad (2)$$

When visual modalities are not available, the situation is different. Other communication channels that are available to convey Mathematical contents (audio and tactile) do not allow a person to get a rapid overview. Indeed the representations used in both cases (Speech and Braille) are intrinsically linear, which means that formulas need to be linearised (3). In most cases this linearisation generates a much longer representation, which is more difficult to understand than the graphical one. For instance in this very simple example the linear version (3) necessitates 11 symbols while the graphical one (2) requires only 7.

$$(x+1)/(x-1) \quad (3)$$

The number of Braille symbols is also fairly limited: there are 6 dots available which can be combined into a maximum of 64 different patterns. Multiple Braille characters are therefore needed to code most Mathematical symbols. For instance, digits 1 to 9 are usually coded with the same Braille patterns as the first 9 letters. Prefixes then indicate whether it should be read as a digit, a lowercase Roman letter, an uppercase Roman letter, a Greek letter, etc...

To reduce the length of formulas, that is to reduce the number of characters and so facilitate understanding, Braille Mathematical notations use complex strategies. For instance, in the British code, the prefix is omitted in the case of a symbol which cannot be a number (after “j”), and lowercase Roman is assumed. In Marburg (German code), the prefix is used only the first time, like a switch, indicating that any other instance of the same pattern will be of the same type. In French and Nemeth (American code) the most frequent case is always assumed (lowercase Roman), and there is a prefix before each other (upper case, Greek, etc.). There is also a special way to represent digits: adding the dot ’6’ to the corresponding letter in French, or, in Nemeth, writing the same pattern on the lower part of the Braille cell (since those symbols do not use the 2 lower dots).

In the case of fractions, block markers identify the numerator and the denominator, making it necessary to reach the fraction symbol to determine that the expression is in fact a fraction. In Italian, the numerator and the denominator markers are not the same and there is no fraction symbol. Then when the first block marker is read, the user immediately knows it is a fraction. In the same kind of idea, Nemeth uses 3 Braille characters: the beginning of fraction, the fraction bar and the end of fraction.

To further complicate things, these Braille Mathematical notations have been developed in different countries, according to the linguistic and cultural history of these countries. Therefore, while the mainstream (visual) representation of formulas is identical in every language, the same is not true for Braille notations. Indeed each Braille Mathematical notation is widely used in its zone of linguistic influence, while it is completely unknown in other countries. In other words, a Braille formula written using the British notation is not understandable by a German speaking reader. This problem is quite important since the number of available Braille documents is very small compared to the number of ordinary Maths books.

2. State of the Art

During the last 2 decades a number of research projects proposed some partial solutions to this problem. First a list of projects focusing on access to Mathematical literature and preparation of Mathematical information have been developed. Most of these projects aim at converting mainstream formats to Mathematical Braille. These converters are used for different purposes. The main is to facilitate the production of scientific Braille documents. For instance they allow a teacher to use a document that was prepared for mainstream students and to convert it into Braille.

Labradoor [1] (LAtex to BRAille DOOR) converts a full LaTeX document including Mathematical formulas into Marburg Braille or into HRTEX (Human Readable TeX). Then several projects have been developed to convert MathML formulas: for instance Bramanet [2] produces French Mathematical Braille, math2braille [3] produces the Braille code in use in the Netherlands, and [4] produces Nemeth. As many Braille Mathematical codes exist, it seems interesting to create a programming library which is able to handle various Mathematical codes, as well mainstream as Braille, in a unified way, so applications using it could propose the best Maths code for each user. MMBT (Multi-Language Mathematical Braille Translator) [5] was a first attempt in this direction. It supported transcriptions from and to LaTeX, MathML, French (revisions 1971 and 2001), British and Italian Braille notations. MMBT has been developed in Java, it was discontinued and is now replaced by UMCL.

UMCL [6] (Universal Maths Conversion Library) is a programming library encapsulating various converters for different Braille codes in a single library, usable through a simple and unique API. UMCL is an open-source project and is portable. It was developed in standard "C" with wrappers to different other programming languages. To make this possible without increasing the complexity, an architecture based on a MathML as central representation of the formula was developed. Currently output modules have been developed for the French notations (revisions 1971 and 2001) and Italian. Beta versions of Marburg and British code are also already available. Input modules for LaTeX and Marburg Braille are also under development.

Another project which helps to produce documents usable by visually impaired people is Infty [7]. It is a large project aiming at giving access to printed Mathematical content. It is based on a core module (InftyReader) which is an OCR specialised in Mathematical documents. It produces a topological representation of the formulas in an XML format. Then this representation can be converted into various formats: MathML, LATEX, HTML, HRTEX, KAMS, and into Unified Braille Code (English) and Japanese Braille code.

In the other directions, converters allow sighted people to access documents written by blind people. This is more difficult to process since Braille codes are context sensitive. INSIGHT [8] proposes a complete system to translate Maths documents with mixed Grade II Braille text and Nemeth code to LATEX. The system processes an image of a Braille sheet (for instance a scanned page) and recognises the Braille dots to produce an ASCII Braille file. Text and Nemeth code are automatically identified and separated to be separately translated. Finally a single LATEX document is produced to be read by a sighted individual.

Several projects [9], [10], [11] focus on better presenting Mathematical contents in speech.

Now we need some new software tools that support the work of blind users, facilitating their understanding and helping them to carry out calculations, while facilitating inclusion in mainstream environment. Indeed more and more such pupils attend to mainstream schools, so it is necessary that these tools are usable with teachers who do not have a specific knowledge of Braille. A few projects have been carried out since a few years.

The Maths Genie [12] is a formula browser that facilitates understanding of formulas using voice. It has been designed to convey the structure of the Mathematical expression as well as its contents. The graphical rendering is synchronised to the audio. The current version supports English, French and Spanish for speech, and offers facilities to add any local language provided that a speech synthesiser is available with the requested interface. A Braille output supporting Nemeth code, based on [4], is available.

Lambda [13] is a Mathematical reading and writing system designed for blind students. The main characteristic of the Lambda project is that it is built on a brand new code. This code is an XML code specifically designed for supporting the Braille transcription into 8-dot pattern national codes. Each Lambda national code has the lambda structure and a Braille character dictionary as close as possible to the official national code. Lambda comes with a dedicated editor, which includes navigation support, input functions (keyboard shortcuts, menus, tool bar). It outputs Lambda Braille, speech synthesis (Mathematical symbols are verbalised in a descriptive language), a visual presentation in a linear code (a specific font in which each Braille character is represented by a visual symbol). A graphical rendering can be displayed on demand (but it is not synchronous to input).

3. Strategies to overcome the problem

The strategies presented here are implemented in MaWEn prototypes that are developed conjointly by Johannes Kepler Universität Linz and l'Université Pierre et Marie Curie within the framework of the MICOLE project. These prototypes are preliminary studies for the development of a Mathematical Working Environment dedicated for blind pupils and students.

3.1 To support collaborative work by synchronising views

One essential idea is that new tools should support collaborative work between blind and sighted individuals, most typically in a mainstream teaching environment, where a single blind pupil needs to be able to collaborate with a sighted teacher and possibly several sighted school mates.

This requires synchronisation of 2 representations using 2 different modalities, one for the blind and one for the sighted. Documents are composite: a main textual structured content includes Mathematical expressions. In MaWEn it is not yet possible to include other kinds of objects (like images with alternative content) but this possibility will be explored if the need appears from the users, for instance in the case of reading a schoolbook. Synchronisation of textual contents in both modalities is not particularly difficult, nevertheless care should be taken that the part which is displayed in Braille always appears on the screen and is clearly identifiable so that sighted users can easily follow the work of the blind pupil.

In the case of Mathematical expressions it is necessary that each of the representations synchronised needs to be a natural representation; that is the representation the readers are used to. In the case of sighted people it needs to be the natural graphical view, while In the case of Blind readers it has to be the Braille Mathematical notation they have been taught, that is the official Braille notation in use in their environment.

Synchronisation must allow each person to point at a location on the document, in the textual part as well as in a Mathematical expression, to show it to the other, in order to highlight an idea or to explain an error. With a graphical view this pointing should be done using the mouse by clicking on the desired location. The specified location would then be highlighted on the alternative view. In the other mode the Blind user must be able to make a selected location display with a different background on the screen.

The synchronisation must also support selection. The current selection must appear clearly as well on the screen as on the alternative display (Braille). This is achieved by showing the current selection with a different background colour. On the Braille bar it is possible to underline the selection using dots 7 and 8.

Synchronisation seems essential in inclusive education. Once again it is not a particular difficulty in the case of textual contents. For Mathematical expressions, it is made possible by the use of MathML conjointly with MathML to Braille converters. Indeed the MathML can easily be displayed graphically thanks to existing software. MathML to Braille converters enable the user to access in real time to a Braille transcription of the formula displayed on the screen. We have developed a model allowing to keep the Mathematical elements described in MathML and the corresponding Braille symbols linked. This model is implemented in the UMCL conversion library and allows MaWEn prototypes to support full synchronisation between the 2 views.

Remark: in the following sections we will only focus on the Mathematical expressions.

3.2 Collapse and Expand sub-branches of expressions

The main obstacle for a blind person to read and understand a Mathematical expression is the length of formulas and the complexity of notations. To get an idea about these crucial problems, and to explain the concept of collapse/expand which may help to overcome them, we would like to talk on the **structural tree** of a formula. As an example, let us consider this equation:

$$L_1 = L_0 . \sqrt{1 - \frac{v^2}{c^2}} \quad (4)$$

It is an equation, with two sides: A “simple expression”, L_1 and a product, composed of another simple expression, L_0 , and a square root. The square root, in turn, is the difference between the number 1 and a fraction. The numerator of that fraction is v^2 , the denominator is c^2 .

This little discussion shows that our formula can be viewed as a tree; it is this very tree that makes up its Mathematical meaning, and that is understood by a sighted person at one glance, even if that person be not a Mathematician at all, or if that formula were even much more complex than this relatively simple example. On the other hand, the blind student will have considerable difficulty understanding the tree, and these difficulties are for sure more than proportional to the length of the formula. The blind person will have to do quite much reading forward and backward in order to finally understand this tree.

It is the idea of collapse/expand to assist the blind person in understanding the tree by presenting certain parts of it in full, while others are presented only in terms of blocks, informing the reader that more detailed information about the branch is available on request.

The above example formula (4) collapsed to the maximum possible, would be represented just by a block. Expanding this to one level, we get:

$$L_1 = L_0 . \text{<block>} \quad (5)$$

Expanding the block will finally give its content, namely:

$$L_1 = L_0 . \sqrt{1 + \text{<block>}} \quad (6)$$

When looking at the above tree expansions, it will be noticed that there are “jumps” in the expansion: For example, when expanding the right side of the equation, strictly following the tree structure would result in a block for the multiplication: $L_1 = \text{<block>}$, and then it would be necessary to expand this block to obtain the expression showed in (5). Also, expanding the square root should yield: $L_1 = L_0 . \sqrt{\text{<block>}}$. We did not propose to follow the structure in such a strict way because we think that this would make the process of expanding clumsy and, hence, less useful: sub-expressions of sufficient simplicity and brevity, like the left factor of the right side of our equation or the contents of the square root, should be displayed at once, without having to go through the strict expand process. It is considered an important task for the developer of a good collapse/expand algorithm to be “judicious enough” to keep these ergonomic aspects in mind.

This collapse/expand is not a new technique: It is used in “Integrated Development Environments” for quite a long time. It is implemented in projects to support blind persons in dealing with Mathematics, e.g., Maths Genie [12] and Lambda [13].

In the MaWEn prototype, we give headlines to the different blocks in order to make it easier to the user to understand the underlying structure. These headlines are mnemonic letters. To come back to our example, the formula, collapsed to the maximum possible, would be represented by the letter E, for “Equation”. Expanding this to one level, the square root is represented by the letter “Q”, so we would get:

$$L_1 = L_0 . Q \quad (5')$$

When expanding the Q, its content would be displayed using the letter “F” for fraction:

$$L_1 = L_0 . \sqrt{1 + \text{<block>}} \quad (6')$$

When looking at the above mnemonics like “E”, “Q”, “F” etc. one might ask how to distinguish them from ordinary Mathematical symbols. Indeed, a possibility of clearly distinguishing such mnemonics from

Mathematical characters must be found. As far as 6-dot Braille representations are concerned, this problem will not occur, because we shall use the mnemonics with dot 7. In case of an ASCII based Maths notation, however, other techniques need to be applied.

Another important requirement for collapse/expand support is the ability to synchronize it with the visual view. This means that, in a collaborative setting of a blind and a sighted person, a means must be found to make collapse of part of a formula visible to the sighted partner, typically, the teacher.

3.3 Editing functions

We do not consider it essential to use Braille Mathematical codes for inputting formulas. The main reason is that converters from Mathematical Braille to MathML are not yet available in the languages we need. It is also necessary to implement a solution which does not imply the use of a Braille keyboard, which is not available in every situation. Anyway as soon as such Braille to MathML converters will be efficiently available in UMCL, it will not be difficult to add this feature.

We propose instead a combined input scheme, where simple expressions such as numbers, variables, or polynomials can be input directly from the Braille or ASCII keyboard, while complex structures like roots, sub and superscripts, fractions etc. are to be input through commands (available in a menu and as keyboard shortcuts)

- **Numbers**: Ordinary Arabic numbers should be input directly, either through a Braille keyboard or through an ordinary qwerty keyboard. In the latter case, the number sign, which is common in many Braille codes, will be input automatically. As an alternative, input through a command should also be possible – this will prove helpful, e.g., in the rare case where Roman numbers are to be input through a standard keyboard. In any case, number input will be terminated by a space. Some caution will have to be applied when deleting or inserting digits: When a digit following the number sign is deleted, the number sign should be removed automatically provided there is no digit left. When inserting a digit it should be checked if another digit is present just after in order not to add a number sign.

- **Variables**: The same philosophy as for numbers will be applied. Special attention deserves the case where a variable consists of more than one letter. We can offer several strategies to handle this: In any case, we propose that letters input directly through the keyboard should be considered distinct one-letter variables, provided that they are not separated by spaces. Now in order to handle the case of multi-letter variables, one could either implement a command with a template (see below), or one might use spaces as separators, as is common with the well-known computer algebra system Mathematica.

- **Fences** (parentheses etc.): These symbols should be input directly through the keyboard.

- **Binary operators** (addition, multiplication etc.): The normal way of input should be via keyboard. However, commands may also be provided, creating templates like this: When the command for “Addition” is issued, an item like “placeholder + placeholder” might appear on the screen, where “placeholder” is a short symbol that can be easily recognized but still not easily confused with true Mathematical characters. For six-dot Braille codes, a double full six-dot cell would be an option. When an expression is input at the spot where the placeholder appears, then the placeholder will be replaced by that expression automatically.

- **Complex structures**: Complex structures such as sub and superscripts, roots, fractions etc. could be input through commands generating templates like in the previous example. An exception could be simple sub or superscripts, which might be input directly using the characters _ (underscore, for subscript) and ^ (caret for superscript), known from the TeX system.

To supply efficient mechanisms of selecting sub-expressions of a formula is very important, first because it is needed to realize collapse/expand support (see previous section), second, because a blind user needs to be able to select a portion of a formula in order to make it visible to his/her sighted partners.

The selection mechanisms are designed with the tree structure of an expression, as discussed in the previous section, in mind: This is because navigating through an expression will need that structure in any case. Pointing to a spot in a formula with the Braille display will select that node in the tree corresponding to the spot. Double clicking at a selected spot will select its parent node - through iterated application of that method, more and more of the expression, up to the whole one, may be selected.

There are some special cases to take care of: since not every character within a formula written in a Braille notation corresponds to a spot in its visual rendering, we need to provide for conventions on what should be selected when such a spot is pointed at on the Braille display. An example of such a character would be the opening and closing sign for a fraction, or the sign that announces the beginning of a sub or superscript. As a general rule, we propose that, when the user points at such a character, then the structure which is announced, or terminated, by that symbol will be selected. In particular, pointing at the “Begin of Fraction” or “End of Fraction” indicator will select the whole fraction, pointing at the symbol announcing a sub- or superscript will select the sub or superscript, pointing at the symbol announcing a root will select the contents of the root.

3.4 Manipulation support functions

Actually carrying out Mathematical calculations is even more difficult than reading and writing formulas. The problems in doing formal manipulations arise because of the complex structures that deploy during a calculation: sighted people may organise a computation such that it can be easily surveyed and navigated, and they use additional graphics to help their reasoning (see equation 1). Then there is a need for powerful editing tools that pupils can use to do calculations more easily. These tools should provide support with respect to the Braille notation itself and not to the Mathematical content.

We would like to develop our ideas on supporting blind persons in doing Mathematics by an example. Consider this exercise of multiplying two sums in parentheses:

$$(3a-5b+6c).(4a+b-3c) \quad (7)$$

The difficulties arising for a blind pupil already with such a relatively simple task were extensively analysed in [14]. Here, we shall describe how those difficulties might be reduced by implementing what we call a *Manipulation Wizard*.

To solve the exercise requires two steps: First the product must be expanded, meaning that every term of the left pair of parentheses has to be multiplied against every term of the right pair. Second, the resulting sum has to be simplified. We shall describe two wizards, one for expanding, and one for simplification, which in our example will be executed in sequence.

We begin with the expansion wizard: Every term of the left factor has to be multiplied by every term of the right one, the order being completely arbitrary. A wizard should address this by offering the various sub-multiplications to the student – (s)he is presented the factors, and needs just to enter the sub-products. By pressing the Down arrow key, the next sub-product should be automatically presented. For every sub-product, there should be an edit field to receive the calculated value from the user, e.g., in the line:

$$3a*4a=... \quad (8)$$

There is an empty box after the equals sign, to receive the result $12a^2$ from the user. The wizard should collect all these results of sub-products, yielding a sum like this:

$$12a^2-20ab+24ac+3ab-5b^2+6bc-9ac+15bc-18c^2 \quad (9)$$

This sum is not yet the final result – it needs to be simplified. For this task, a *Simplification Wizard* should be designed: to simplify an algebraic sum, the following systematic procedure can be applied: Iterate through all the terms of the sum, choosing a "reference term" – this is the outer iteration. Compare the reference term to all terms coming after it, in order to find terms to be contracted – this is the inner iteration. If the reference term can be contracted with subsequent terms, collect the contractions for the resulting sum. If you do not find any successors of a reference term to be contracted with it, take the reference term into the result unchanged.

Even when strictly following this scheme, you may fall into two difficulties: First, you shall find terms succeeding a reference term that already were compared to a prior reference term, such that they do not need consideration anymore. Second, the procedure may let you view terms as reference terms which were successors of older reference terms, such that they also were already treated and are now to be skipped.

The *Simplify Wizard* addresses these difficulties by quite a simple procedure: As always, it presents you the various pairs "reference term – successor" beside each other. You may iterate through the successors with a pair of keys, e.g., Page-Up and Page-Down, leaving the reference term unchanged. You may take the next reference term with another key, e.g., the End key.

When a particular reference term is chosen, you may mark it with Control-M – this sets an opening square bracket left to it, which will remain adherent, such that, when you come across that term in the future, you will see that it was already considered. Also, it will be copied into a temporary edit buffer, allowing you to contract successors with it. Equally, when you decide a particular successor to be contracted to the reference term, then you should mark it with Control-M: This also makes an opening square bracket adhere to it, such that it can be seen as already having been used in the future; moreover, it copies the successor into the temporary buffer, to facilitate computing the contracted term.

Once a particular reference term has been fully processed, i.e., once all its successors have been considered, the sum representing the contraction, or the unchanged reference term, can be written into the temporary buffer. Once the various terms having formed the contracted sum were deleted manually, you may press the Down arrow key in order to copy the interim result into the final one, automatically clearing the temporary buffer to give room for the next reference term.

Although wizards of that kind would reduce much of the tremendous difficulties encountered by blind persons in doing Mathematics, it was argued that they are “too supportive”, which means that they take so much Mathematical work away from the pupil that (s)he might fail to understand the procedures of calculation. It is indeed an open research question how to develop software routines which, although being supportive, still leave enough responsibility on behalf of the pupil. As an example, the *Expansion Wizard* might be modified such that the pupil may carry out the iterations through the terms of the two factors by him/herself: By walking through one factor, (s(he may choose one of its terms as reference term by selecting it. In a second step, this reference term should be made constantly accessible to the pupil, while (s)he walks through the second factor in order to do the inner iteration by selecting terms of it. Once the result of a sub-product is computed, the student may issue a command to copy the result to a final sum, which (s)he may inspect after all the iterations are completed.

The ideal case would be to have a kind of “Method Base”, consisting of elementary support functions, from which more complex supportive wizards similar to the ones sketched here should be modelled through a configuration language, or, perhaps, through a wizard, in turn. Such a meta-tool should be made accessible to the teacher, or perhaps to a very advanced pupil, in order to tailor support to the pupil’s Mathematical maturity and abilities.

3.5 Context sensitive help for Braille codes

Additionally there is also a tremendous need for providing contextual support on the Braille Mathematical code. Braille Maths codes are normally hard to learn for a blind pupil, and almost impossible to learn for a sighted teacher. In order to support both the pupil and the interested teacher, a tool to furnish context-sensitive help for a Mathematical text written in a Braille notation would be desirable. Such a tool might display information about a character read by the Braille display, information about its meaning in Mathematical context for the blind partner, and, if possible, the black-print analogue of the character for the sighted partner. It should also offer concrete examples of its use, and an account of the important rules to be followed when using it.

4. Conclusion

We believe that the strategies described in this paper will effectively support the work of Blind individuals. They will be implemented in MaWEn and evaluated in school situations. One common thread with the other projects cited in the state of the art is the use of MathML. It is of tremendous importance to generalise the use of MathML in mainstream scientific documents so these documents can be easily accessed with those specialised tools.

Acknowledgements

The MaWEn prototypes are developed within the framework of the MICOLE project, which is funded by the INFSO DG of the European commission, within the framework of the IST 2002 programme (FP6/IST/eInclusion) under the reference IST-2003-511592 STP. This paper was written with the support of the @Science project which co-funded by the INFSO DG of the European Commission4, within the eContentPlus programme (FP6/thematic network priority), under the reference ECP-2005-CULT-038137. The contents of this paper is the sole responsibility of the authors and in no way represents the views of the European Commission or its services

References

[1] Batušic M., Miesenberger, K., and Stöger B. (1998), “Labradoor, a contribution to making Mathematics Accessible for the Blind”, In Edwards, A., Arato, A., and Zagler, W., editors, Proc. ICCHP 98 (6th International Conference on Computers Helping People with Special Needs), Oldenbourg, Wien, München.

[2] Schwebel, F. (2003). "BraMaNet logiciel de traduction des mathématiques en braille." [http://handy.univ-lyon1.fr/projets/bramanet/].

[3] Crombie, D., Lenoir, R., McKenzie, N., and Barker, A. (2004). "math2braille: Opening access to Mathematics." In Miesenberger, K., Klaus, J., Zagler, W., and Burger, D., editors, Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), volume 3118 of LNCS, pages 670-677, Berlin. Springer.

[4] Stanley, P. B. and Karshmer, A. I. (2006). "Translating MathML into Nemeth Braille Code." In Miesenberger, K., Klaus, J., Zagler, W., and Karshmer, A., editors, Proc. ICCHP 2006 (10th International Conference on Computers Helping People with Special Needs), volume 4061 of LNCS, pages 1175-1182, Linz, Austria. Springer.

[5] Moço, V. and Archambault, D. (2003). "Automatic Translator for Mathematical Braille." In Stephanidis, C., editor, Universal Access in HCI - Inclusive Design in the Information Society, volume 4, pages 1335 - 1339, Mahwah, New Jersey, USA. Lea.

[6] Archambault, D., Fitzpatrick, D., Gupta, G., Karshmer, A., Miesenberger, K., and Pontelli, E. (2004). "Towards a Universal Maths Conversion Library." In Miesenberger, K., Klaus, J., Zagler, W., and Burger, D., editors, Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), volume 3118 of LNCS, pages 664 - 669, Berlin. Springer.

[7] Suzuki, M., Kanahori, T., Ohtake, N., and Yamaguchi, K. (2004). "An Integrated OCR Software for Mathematical Documents and Its Output with Accessibility." In Miesenberger, K., Klaus, J., Zagler, W., and Burger, D., editors, Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), volume 3118 of LNCS, pages 648-655, Berlin. Springer.

[8] Annamalai, A., Gopal, D., Gupta, G., Guo, H., and Karshmer, A. (2003). "INSIGHT: a Comprehensive System for Converting Braille based Mathematical Documents to Latex." In Stephanidis, C., editor, Universal Access in HCI - Inclusive Design in the Information Society, volume 4, pages 1245 - 1249, Mahwah, New Jersey, USA. Lea.

[9] Raman T.V. (1994), “Audio Systems for Technical Reading”, PhD thesis, Department of Computer Science, Cornell University, NY, USA.

[10] Stevens R.D. (1996), “Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People”, PhD thesis, University of York, UK.

[11] Fitzpatrick D. (2006), “Mathematics: How and What to Speak”, In Miesenberger, K., Klaus, J., Zagler, W., and Karshmer, A., editors, Proc. ICCHP 2006 (10th International Conference on Computers Helping People with Special Needs), volume 4061 of LNCS, pages 1199-1206, Linz, Austria. Springer.

[12] Karshmer A., Gupta G. and Gillan D. (2002), "Architecting an Auditory Browser for Navigating Mathematical Expressions", In K. Miesenberger, J. Klaus, and W. Zagler, editors, Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs), volume 2398 of LNCS, pages 477-485, Linz, Austria. Springer.

[13] Schweikhardt, W., Bernareggi, C., Jessel, N., Encelle, B., and Gut, M. (2006). "LAMBDA: a European System to Access Mathematics with Braille and Audio Synthesis.", In Miesenberger, K., Klaus, J., Zagler, W., and Karshmer, A., editors, Proc. ICCHP 2006 (10th International Conference on Computers Helping People with Special Needs), volume 4061 of LNCS, pages 1223-1230, Linz, Austria. Springer.

[14] Stöger B., Batušic M. and Miesenberger K. (2004), "Mathematical Working Environment for the Blind Motivation and Basic Ideas" In Miesenberger, K., Klaus, J., Zagler, W., and Burger, D., editors, Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), volume 3118 of LNCS, pages 656-663, Berlin. Springer.

2.2 [HCI'2005a]

Titre	**Multimodal interaction with mathematical formulae : Access, communication and support for blind people in doing mathematics**
Auteur(s)	Bernhard Stöger[a], Klaus Miesenberger[a], Mario Batušić[a] et Dominique Archambault
Affiliations	[a]Johannes Kepler Universität Linz, Autriche
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of HCI International 2005 Conference (11th International Conference on Human-Computer Interaction)
Editeur(s)	Constantine Stephanidis
Lieu/Date	Las Vegas, Nevada, États-Unis, Juillet 2005
Pages	7 pages (proceedings on CD-Rom)

Voir mémoire section **3.1.2.2**, page 28

Multimodal Interaction with Mathematical Formulae: Access, Communication and Support for Blind People in Doing Mathematics

Bernhard Stöger, Klaus Miesenberger*, Mario Batusic*, Dominique Archambault***

* Universität Linz, Institut "Integriert Studieren"
Altenbergerstrasse 69, A-4040 Linz, Austria
bernhard.stoeger |klaus.miesenbergerg|mario.batusic@jku.at

** Université Pierre et Marie Curie
Inserm U483/inova – 9, quai Saint Bernard – 75252 Paris cedex 5 – France
Dominique.Archambault@snv.jussieu.fr

Abstract

Modern Information and Communication Technology (ICT) has changed our style in getting access and also in performing things. In terms of education we have learned to use ICT as a means for reading, writing, communicating, searching, arranging and solving problems.
People with disabilities and in particular blind people benefit a lot from these new possibilities as they can make use of assistive technologies (AT) to get access to ICT and thereby work integrated on the same sources of information as anybody else.
Doing mathematics, solving mathematical problems is one area where ICT and also AT are not employed in the same way as for other areas. Doing mathematics is still very much "paper based". ICT and AT is used intensively to get access, to read and also to write mathematics. Communicating mathematics between sighted and blind people and especially doing mathematics with a computer – solving more complex problems – is still an area asking for increased attention. Blind people are still at a disadvantage in this area.
As there is not really a working environment for doing mathematics for sighted people we are still lacking an equivalent alternative for blind people. Of course there are a lot of facilities for writing, calculating, … but the process of doing mathematics is still based on paper and pencil.
In the EU funded MICOLE project (Multimodal collaboration environment for inclusion of visually impaired children) we want to address this issue by having a closer look at the way how sighted children and youngsters learn to do mathematics. We want to make the methodologies of sighted people in doing mathematics with paper and pencil which are implicit to the visual style and arrangements on paper explicit. Based on these concepts we want to develop computer supported functionalities for blind people offering similar support to the blind as sighted have when doing mathematics with paper an pencil. Having an in depth look into mathematical school books will help us to find these functionalities and to define equivalent alternatives for Braille or speech based interfaces.

1 Introduction

The past three decades saw considerable progress in access to information for the group of blind and visually impaired people: Thanks to modern information technology in the mainstream and to very specialized adaptive and assistive technologies, blind and visually impaired people are now able to deal independently and efficiently with almost every piece of information that is composed of pure text. Despite current strong trends towards graphical presentation, text still covers the majority of relevant contents for private and professional life, such that information access for the target group is currently accomplished to a very large extent.
Despite intensive research efforts carried out over the last years, blind and visually impaired people are still excluded from an efficient usage and handling of graphical contents. Since Mathematics is presented in a highly graphical way most of the time, this exclusion implies considerable restrictions in access to Mathematics, too. Although "accessibility" is put in place, "usability" and especially support functionalities in "doing" mathematics are very low. This paper analyses the major issues, outlines the existing approaches to a possible solution and describes the current activities within the MICOLE project towards a comprehensive, multimodal software answer to this problem.

2 State of the Art

The problems faced by the target group with respect to Mathematics fall into four basic categories (Miesenberger, Batusic & Stöger 2003):

1. Access to mathematical literature (books, teaching materials, papers etc.)
2. Preparation of mathematical information (presenting school exercises, writing papers etc.)
3. Navigation in mathematical expressions and communication between blind and sighted people
4. Doing Mathematics (carrying out calculations and computations at all levels, doing formal manipulation, solving exercises)

While the first two problems have been addressed to produce solutions, which are at least satisfactory to some extent, and basic approaches are available to the third problem, almost nothing was achieved by now to support the target group in solving tasks of the last category (Guo, Karshmer, Weaver, Mendez & Geiger 2000, Karshmer 2002).
The third category, navigation within mathematical contents, deserves attention because it is much more complex than just reading them like ordinary textual information: For sequential reading, as sufficient with text, will not give one the understanding of a complex mathematical expression. To grasp the meaning of a formula, one needs to repeatedly scan passages of it, and to jump over irrelevant portions, which is a heavy challenge if you cannot see. A supportive software environment, which, as a minimum requirement, should furnish collapse and expand functionality, would be needed. (Karshmer, Gupta, Miesenberger, Pontelli & Guo 2001)
An additional challenge is the communication problem between blind and sighted people. The visual manner of displaying formulae differs a lot from a Braille or speech based structure. Due to this communicating, at least synchronising where and what one reads/works on at a certain moment is still an unsolved problem.
What was said about navigation in formulae is true, but to a much larger extent, for the last category, the task of actually doing Mathematics. Even at secondary school level, formulae tend to deploy to high complexity quite soon within a computation what makes tasks like maintaining an overview, finding relevant spots, minimizing errors in remembering and copying. All these are tasks non-trivial for sighted people, thus still much more challenging for blind and visually impaired students and mathematicians. Even blind people who successfully mastered high-level mathematical courses complain about the cognitive overload, from which it becomes clear that it is a pressing issue for mathematical education of the blind and visually impaired.

The above mentioned software solution for the support of navigation within mathematical expressions needs a substantial extension to support the much more complex task of effectively carrying out computations on all levels for the target group. Such a software suite, which we envisage and which combines the already attained solutions for the first three categories of tasks with new solutions for the latter categories, will be referred to as a Mathematical Working Environment (MaWEn). The following sections shall

- briefly sketch the activities by now undertaken to solve the problems of the first three categories,
- present a concrete example to motivate the need of a software solution addressing the problems of navigation and manipulation,
- list requirements to be fulfilled by a comprehensive Mathematical Working Environment and
- finally sketch a raw draft of modules to be implemented in order to construct such an environment.

3 Contributions to overcome the problem

For the last 10 years the above mentioned issues were envisaged, and research was started. We developped first a converter from LATEX into Marburg mathematical Braille Notation – the schoolbook preparation system LaBraDoor (Batusic, Miesenberger & Stöger 1998). This system is in use in Austria for the schoolbook preparation on all school levels. That project will mainly deliver the know-how for the import module of MaWEn.

As the next step in the overall problem solving process we developed a backtranslator from Marburg Mathematical Notation into MathML (Miesenberger, Batusic & Stöger 2003). This was accomplished within the framework of the international project UMA (Universal Math Access) (Universal Math Access Project). As the most important outcome of this project an international Group for Universal Mathematics Access was initiated. The first common achievment of the Group will be an universal math converter library for interconverting between various standard and braille mathematical formats such as:

- MathML
- LATEX
- Nemeth Maths Code (USA and Canada)
- Marburg Mathematical Braille Notation (German speaking area)
- French Braille Maths Code

The system allows an easy integration of additional notations.

4 Support Functions For Data Manipulation: A Case Study

We would like to use a concrete example to describe the sketched difficulties in formal manipulation faced by blind people. Consider the following simple exercise:

$$(1)\ (3a - 5b + 6c) . (4a + b - 3c)$$

Let us first describe the process, independently of visual impairment:

1. Each summand of the right parenthesis is multiplied by each summand of the left one, giving a long sum that has to be simplified in a second step. In detail:

(a) The term 4a is multiplied by all the summands of the left parenthesis consecutively, giving the sum:

$$(2)\ 12a^2 - 20ab + 24ac$$

(b) The same is done with the expression +b, giving:

$$(3)\ +3ab - 5b^2 + 6bc$$

(c) Finally, the term -3c is multiplied against the terms in the left parenthesis, giving

$$(4)\ -9ac + 15bc - 18c^2$$

2. The three resulting sums are consticted to the expression

$$(5)\ 12a^2 - 20ab + 24ac + 3ab - 5b^2 + 6bc - 9ac + 15bc - 18c^2$$

3. the next step is the simplification:

The sum is scanned from left to right in order to find terms with the same sequence of letters; if found, the terms will be constricted. In detail:

(a) The term 12a2 is not matched by any other term, such that it will be copied “as is” into the final result.
(b) The term -20ab has to be constricted with +3ab – the addition yields -17ab
(c) The term +24ac unites with -9ac to +15ac
(d) The term -2b2 does not have a counterpart, such that it can be simply appended to the final sum.
(e) The term +6bc adds to +15bc giving +21bc
(f) The term -18c2 is a singleton, concluding the sum.
(g) We thus obtain

$$(6)\ 12a^2 - 17ab + 15ac - 5b^2 + 21bc - 18c^2$$

as the simplified final result of the calculation.

When a blind student solves an exercise like this via braille display and computer in the conventional way, one encounters a number of difficulties. In order to understand these problems, let us take a look at the way how a blind student would carry out the steps detailed above. For first one will move the exercise to a free place in a text document, either by manual copying or by cut and paste. One will open a new line, beginning with an equals sign, to enter the first step, namely the factorisation of the two parentheses. Now, the right parenthesis has to be scanned summand by summand. To this end, the student has to direct the braille display to the exercise, to search the second factor, and to find (and remember) its first summand. One will do his/her best to remember that expression well, because it is needed as the reference term to be multiplied against all the summands of the leftmost parenthesis.

Once the first summand is found, the product is calculated and committed to memory, since it needs to be inserted into the interim result after the equals sign. This, in turn, requires a jump to the equals sign.

In order to find the second reference term, another jump into the exercise is needed; furthermore, the student has to know that the first reference term was already processed, which challenges his/her memory: Not within this initial stage where only two reference terms were processed, of course, but in the course of the calculation it can easily happen that the position within the second factor may be forgotten; to synchronize in such an event would mean an investigation of the steps carried out so far, involving a jump to the interim result.

Already with this first step in the calculation - and this is the easier one - two fundamental difficulties become apparent:

1. One has to jump constantly between the exercise and the interim result, namely, for every partial product one jump to the exercise in order to calculate the partial product, and one jump back to the interim result in order to append the partial product. With 9 partial products, as in our example, this sums up to 18 jumps. The jumps into the interim result are simplified by a function of the screen reader that performs a jump to the cursor, because it is expected that, as the interim result develops, the cursor moves along it. However, the jumps into the exercise require navigation with the braille display in any case.
2. Beside the jumps, the task imposes considerable load on the blind student’s memory, namely, in two respects:
    a. One has to well remember the computed partial product to be able to append it to the interim result after having jumped to its end.
    b. In order to compute the next partial product one needs to know what was already processed. This results in the need to remember two positions, one within the left and one within the right factor.

Step 2, the simplification of the computed expanded sum, contains more and greater difficulties: The interim result, which now becomes the exercise, has to be searched summand by summand from left to right, and for every summand it has to be checked whether it can be constricted with other members of the sum. This begins with a jump to the exercise, where the first summand $12a^2$ has to be committed to memory. Now the whole long sum has to be scanned, and every summand needs to be checked to see whether it is a multiple of a^2. In the end the student can determine that this is not the case, such that he/she can append the term $12a^2$ to the final result. A jump to the exercise produces the next reference term $-20ab$. Checking against the whole remaining sum, in whose course the reference term $-20ab$ has to be constantly remembered, yields that it can be constricted with $+3ab$, which produces the term $-17ab$ to be appended to the result.

Continuing, we get the term $+24ac$ to be constricted with $-9ac$ yielding $+13ac$ for the final result.
With the next term, $+3ab$, an additional problem arises: A scan through the sum to the right does not lead to any summand to be constricted with the term. Rather, the sum now has to be searched to the left of the reference term $+3ab$ in order to see that this term was already processed, namely when it was constricted with $-20ab$.
Continuing in this manner, the simplification of the sum can be completed.

By the above analysis, these difficulties were isolated:

1. Constant jumping between exercise and result

2. necessity to remember the positions within the exercise
3. need to commit reference terms to memory
4. load on memory by partial products that have to be appended to the result
5. requirement to search the exercise in every step bidirectionally, both to the right and to the left of the reference term.

The Mathematical Working Environment, MaWEn, shall furnish software tools to support a blind student in tasks like the above. Especially, load on memory should be considerably reduced. Some concrete measures in this direction are listed below:

1. Window oriented navigation: every step in a calculation, the exercise, the interim results and the final result, shall be stored in a separate window, probably a multiline edit field. Simple keyboard commands shall navigate between the various windows.
2. No need for manually appending a partial result to a final one: Once a partial result is to be computed – multiplication of two factors or joining of two summands in our example – the student can open an edit field that will take the partial result inserted by him/her. Once that field is closed, the partial result is automatically appended to the end of the result, and an automatic jump back to the exercise is performed.
3. Controlled jumps: The jump back to the exercise shall be performed such that the position from which processing is to be continued becomes apparent: For our example, in step 1, expansion of the product, both the reference term in the right factor and the next term in the left factor to be multiplied by the reference term will be selected.
4. These measures shall largely eliminate the need to remember positions.
5. Parts that have already been processed, like the term +3ab in step 2, are either not displayed at all, or they are marked as ”already processed”.
6. Direct jumps into the final result during a calculation will be possible, although not necessary: For every partial result inserted into the edit field mentioned under 1. shall be automatically appended to the end of the final result.
7. Shortening of expressions as sighted users do by marking already handled expressions.

It’s quite plain that these and also further supportive functions to be implemented in the MaWEn can be switched on and off. For otherwise a student who is about to learn a particular kind of mathematical exercise would never get into touch with the above outlined difficulties. Rather, the system will be designed more like a ”tool box” such that support for a category of tasks of one level of complexity will be offered only if the student is about to learn the next higher level, or at least if she/he has gained a reasonable understanding of the mentioned level of complexity.

5 Mathematical Working Environment – MaWEn

”Doing” mathematics as a sighted person is based on a lot of skills and knowledge which is implicit to the visual methods of doing math. MaWEn intends

- to find the implicit skills of doing mathematics which are represented in the visual methods (which you find e.g. described as pathways of doing mathematics in schoolbooks),
- to make them explicit and structure them in functionalities which should
- form the basis of a supporting environment and
- allow the communication between sighted and blind people.

Already in 1996 we stated the most important requirements of a future computer aided mathematical working environment for the braille users (Miesenberger, Burger, Batusic & Stöger). These requirements, in addition to those of speech output users which are tackled by our GUMA project partners, didn’t lose anything of their actuality. The main requirements of a Braille user in a mathematical working environment can be enumerated as follows:

- Accessing as many mathematical documents as possible in common formats
- Ability to present all mathematical constructs in Braille
- Using a Braille notation of once own choice

- Parallel Braille and graphical view of the current formula
- Software support in the concrete data manipulation during a “manual” computational task.

The sections up to now described the available converting tools and gave a detailed insight into the support methods for data manipulation beeing currently developed in a PhD thesis. To round off the presentation of MaWEn we bring a short outline of its specific modules planned so far. We don’t refer to the common modules needed in each browser/editor software.

5.1 Overall MaWEn Features

It’s obvious that MaWEn has to fulfil all software accessibility guidelines to guarantee the best possible ergonomics and efficiency (e.g. keyboard bindings for all functionalities, use of exclusively standard Windows libraries/objects for GUI). All modules are concieved as independent building blocks in DLL format which can be used in other applications.

5.2 GUI Module

The GUI module is responsible for the “look and feel” of the main application. Even though the first version of MaWEn should be a MS Windows 2000/XP application, it is planned that all other modules are programmed in such a way that they can interact also with an alternative GUI module made for an other operating system. A speciality of this module is the “dual presentation mode”. It concerns one of the main MaWEn features: displaying the current mathematical formula both in a Braille code and as a 2D visual rendering.

5.3 Data Storage Module

Documents processed within MaWEN shall be saved for further reference: As document formats we envisage a flavour of XML or a mathematical standard such as OpenMath (Open Math Project) or OMDoc (OMDoc Project).

5.4 Data Manipulation and Communication Support Module

This module contains all support functions and data manipulation rules which can be selectively used. In the section 4 we showed by an example how the functions of this module could be helpful in the editing/calculating process.

As mentioned only now the help engine works according to some support rules. An easy-to-use and WYSIWYG tool will be developed to enable users or teachers to design new support rules. For all existing rules the user will have a possibility to set options to enable, disable or to apply them selectively to classes of mathematical expressions. To explain the process of defining support rules would go beyond the scope of this paper. To give roughly a basic understanding of the support rules concept, we can see them somewhat similar to the filtering rules in a mail application. There is an extensible set of appliable actions such as e.g. ”hide item, show item, put item in memory (or bookmark item), mark item as finished. A rule defines an action (or a series of actions) to be executed on items/mathematical objects according to the matching conditions.

6 Acknowledgement

This work was done in the frame of the project ”The Development of a Tool to Enhance Communications between Blind and Sighted Mathematicians, Students and Teachers - a Global Translation Appliance” funded by the US Department for Education. Future work will be funded by the European Commission, DG IST in the frame of the Micole project (511592).

7 References

Batusic, M., Miesenberger K. & Stöger, B.: LABRADOOR, a contribution to making mathematics accessible for the blind, in: Edwards, A., D., N., Arato, A. & Zagler, W., L.: *Computers and Assistive Technology - 6th International Conference on Computers Helping People with Special Needs, ICCHP ´98*, Oldenbourg, Wien, München 1998

Miesenberger, K., Batusic & M., Stöger, B.: Parser for the Marburg Mathematical Braille Notation, in: Stephanidis, C. (ed.): *Universal Access in HCI – Inclusive Design in the Information Society*, Volume 4, Lea, Mahwah, 2003

Karshmer, A.: Access to Mathematics by Blind Students, in: Miesenberger, K., Klaus, J., Zagler, W. (eds): *Computers Helping People with Special Needs - 8th International Conference, ICCHP, Linz*, Austria, July 2002, Proceedings, Springer Heidelberg, 2002

Karshmer, A., Gupta, G., Miesenberger, K., Pontelli, E., Guo, H.: The Development of a Tool to enhance communications between blind and sighted mathematicians, students, and teachers: A Global Translation Appliance, in: Stephanidis, C. (ed.): *Universal Access In HCI - Towards and Information Society for All*, Lawrence Erlbaum, London 2001

Guo, H., Karshmer, A., Weaver, C., Mendez, J., Geiger, S.: Computer Processing of Nemeth Braile Math Code, in: Vollmar, R., Wagner, R.: *Computers Helping People with Special Needs, ICCHP ´00*, Oldenbourg, Wien, München 2000

Miesenberger, K., Burger F., Batusic, M., Stöger, B.: Access to Mathematics for the Blind - Defining HrTEX Standard. *Proceedings of the 5th International Conference on Computers Helpling People with Special Needs*, ICCHP ’96 Linz, Austria, July 1996, pp. 609-616. Oldenbourg Wien München, 1996

Universal Math Access Project: http://karshmer.lklnd.usf.edu/

Open Math Project: http://www.openmath.org

OMDOC Project: http://www.mathweb.org/omdoc

2.3 [ASSETS'2007]

Titre	**A software model to support collaborative mathematical work between Braille and sighted users**
Auteur(s)	Dominique Archambault, Bernhard Stöger[(a)], Mario Batušić[(a)], Claudia Fahrengruber[(a)] et Klaus Miesenberger[(a)]
Affiliations	[(a)]Johannes Kepler Universität Linz, Autriche
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of the ASSETS'2007 Conference (9th International ACM SIGACCESS Conference on Computers and Accessibility), publiés par ACM SIGACCESS
Editeur(s)	Enrico Pontelli et Shari Trewin
Lieu/Date	Tempe, Arizona, États-Unis, Octobre 2007
Pages	115–122

Voir mémoire sections **3.2.1**, page 34
et **3.4.3**, page 44

A Software Model to Support Collaborative Mathematical Work between Braille and Sighted Users

Dominique Archambault
Université Pierre et Marie Curie - Paris 6
9, quai Saint Bernard - B23
75005 Paris, France
dominique.archambault@upmc.fr

Bernhard Stöger
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, Austria
Bernhard.Stoeger@jku.at

Mario Batusic
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, Austria
Mario.Batusic@jku.at

Claudia Fahrengruber
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, Austria
Claudia.Fahrengruber@jku.at

Klaus Miesenberger
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, Austria
Klaus.Miesenberger@jku.at

ABSTRACT

In this paper we describe a software model that we have developed within the framework of the MaWEn project (Mathematical Working Environment). Based on the MathML standard, this model enables collaboration between sighted people and users of Braille. It allows for synchronisation of Braille and graphical views of scientific contents as well as offering improved navigational functions for Braille users, in both reading and editing modes. The UMCL (Universal Maths Conversion Library) is used to support various national Braille Mathematical notations. After presenting the model, its implementation in MaWEn prototypes is described.

Categories and Subject Descriptors

D.2.2 [**Design Tools and Techniques**]: user interfaces

General Terms

Design, Human factors, Standardization

Keywords

Braille, collaborative work, conversion, inclusive education, mathematics

1.INTRODUCTION

The study of Mathematics has always been particularly difficult for blind individuals. Indeed we can observe that a large majority of blind pupils do not succeed in maths studies, while the average mainstream sighted pupil succeeds more readily. As maths is crucial in most science disciplines, this limits study options and future job opportunities for blind people.

We maintain that there is no reason that mathematical semantics can not be understood because of blindness; the biggest barrier is in fact access to mathematical content, which is only possible through speech or Braille.

The MaWEn (*Mathematical Work Environment*) project is a long-term project run by the *Johannes Kepler Universität Linz*, Austria and the *Université Pierre et Marie Curie - Paris 6*, France. The overall goal of MaWEn is to develop a Mathematical environment for education allowing pupils in mainstream schools to access digital scientific contents and to create and edit scientific documents. MaWEn aims to provide synchronised graphical and Braille displays, support functions and on-line help with Braille codes.

This paper focuses on our suggested model which enables support for synchronisation and navigation, as well for reading and editing, and outlines the way in which the model is implemented in MaWEn. First we describe the main problems that blind individuals working with mathematics face and then present a rapid state of the art overview.

2.NON VISUAL REPRESENTATION OF MATHEMATICAL EXPRESSIONS

When visual modalities are not available alternative communication channels must be called on; both audio and tactile channels can be used to convey mathematical contents. The representations used in both cases (speech and Braille respectively) are intrinsically linear and therefore do not allow a person to get a rapid overview. Mathematical expressions need to be linearised. In most cases the linearisation generates a much longer representation which is more difficult to understand than the graphical one.

For instance the following very simple fraction is usually written with a visual layout (1) that allows for very quick understanding of its semantics. The reader realises at first glance that the expression is a fraction and that the numerator and the denominator are rather short. The same expression written in a linear way (2) requires the use of many more symbols (11 compared to 7 for the first), and needs a little more thought to be understood.

$$\frac{x+1}{x-1} \quad (1) \qquad (x+1)/(x-1) \quad (2)$$

From this linear representation it is possible to print a Braille version using specific Braille characters for mathematical symbols (for example here the fraction bar and the operators). As Braille is based on a six-dot pattern it can only form 64 different symbols, which is insufficient to represent the large variety of maths symbols necessary. Consequently most symbols are represented using several Braille characters (in most cases two or three are needed, sometimes more), making the formulas even longer.

Throughout the twentieth century various strategies were developed to reduce the length of Braille formulas, leading to the creation of specific Braille notations for maths. These specific notations are usually context sensitive, enabling significant reductions in the length of formulas. If being converted to various Braille mathematical codes expression (1) would actually require up to 19 Braille characters. Note that this example is quite simple but in the case of complex formulas the number of symbols can increase dramatically.

In the case of speech, the length of expressions is still a problem because the listener has to remember the entire formula. It is not possible to go backwards or forward readily. Speech modality causes further ambiguities. Indeed the previous formula would be naturally spoken like this: "x plus 1 over x minus 1". However, there are three other ways in which this sentence can be understood: see formulas (3), (4), (5).

$$x+\frac{1}{x-1} \quad (3) \qquad x+\frac{1}{x}-1 \quad (4) \qquad \frac{x+1}{x}-1 \quad (5)$$

State of the art overview

Over the last two decades a number of research projects have proposed various partial solutions to this problem. [4] sought to provide a comprehensive state of the art study.

Firstly a number of projects focusing on access to mathematical literature and preparation of mathematical information were developed. Most of these projects aimed at converting mainstream formats to mathematical Braille. These converters are used for different purposes, the main one being to facilitate the production of scientific Braille documents. For instance they allow a teacher to use a document prepared for mainstream students by converting it into Braille.

Labradoor [5] (LAtex to BRAille DOOR) converts a full LaTeX document including mathematical formulas into Marburg Braille or into HRTEX (Human Readable TeX). Several projects were then developed to convert MathML formulas: for instance Bramanet [18] produces French mathematical Braille, math2braille [7] produces the Braille code in use in the Netherlands, and [15] produces Nemeth. The UMA system [11] is a collaboration between several institutions for making mathematics universally accessible. It includes translators to/from several mainstream formats and specific Braille notations and notation-independent tools for aural navigation of mathematics.

As many Braille mathematical codes exist, it seemed useful to create a programming library which is able to handle various mathematical codes, mainstream as well as Braille, in a unified way, so that applications using it could propose the maths code most suitable for each user. MMBT (Multi-Language Mathematical Braille Translator) [12] was a first attempt in this direction. Developed in Java, it has now been discontinued and replaced by UMCL (see below). Several projects [9, 13, 16] focus on better presenting mathematical contents in speech.

In the reverse mode, converters allow sighted people to access documents written by blind people. This is a more difficult processing task since Braille codes are context sensitive. INSIGHT [1] proposes a complete system to translate to LaTeX Maths documents including mixed Grade II Braille text and Nemeth code. The system processes an image of a Braille sheet (for instance a scanned page) and recognises the Braille dots producing an ASCII Braille file.

Other projects aim at supporting the work of blind users, facilitating their understanding and/or helping them to carry out calculations.

The Maths Genie [10] is a formula browser that facilitates understanding of formulas using voice. It was designed to convey the structure of the mathematical expression as well as its contents. The graphical rendering is synchronised to the audio. The current version supports English, French and Spanish for speech and offers feature for adding any local language provided that a speech synthesiser is available with the requested interface. Braille output supporting Nemeth code, based on [15], is available.

Lambda [8] is a mathematical reading and writing system designed for blind students. The main characteristic of the Lambda project is that it is built on a completely new code. This code is an XML-based code specifically designed for supporting the Braille transcription into 8-dot pattern national codes. Each Lambda national code has the lambda structure and a Braille character dictionary resembling as closely as possible the official national code. Lambda comes with a dedicated editor, which includes navigation support and input functions (keyboard shortcuts, menus, tool bar). It outputs Lambda Braille, speech synthesis (mathematical symbols are verbalised in a descriptive language) and a visual presentation in a linear code (a specific font in which each Braille character is represented by a visual symbol). The graphical rendering can be displayed on demand but it is not synchronous to input.

3.THE MAWEN PROJECT

The MaWEn project was started to try to give a response to the problems mentioned above. For this purpose we have been developing a number of prototypes in order to build the underlying model and to evaluate with end users the various features needed.

Two synchronised views

To support inclusive education, it is important to incorporate a feature that allows for two synchronised views of mathematical expressions: one view being specifically for the blind pupil, the other for the sighted teacher and peers.

For the blind pupil's view, it was decided to use Braille and not speech for various reasons. First in a classroom situation the use of sound would be disruptive for the other pupils. In a similar way the use of a headset would isolate the blind pupil from the rest of the classroom. Another reason for using Braille was that editing is particularly difficult with speech synthesis. Furthermore Braille, though one-dimensional, allows for at least a limited spacial overview of a formula whereas speech is completely sequential. Also, the cursor routing buttons which can be found on every modern Braille display are a good means to point accurately at a particular point of an expression. Finally, Braille technology is widely used with pupils, especially in German-speaking and nordic countries.

The view designed for sighted individuals, who do not necessarily understand any Braille mathematical notation, must display a graphical representation of the same formula. It will be referred to as the graphical view throughout the rest of this paper.

These two views have to be synchronised which means firstly that they should be updated simultaneously during editing and that any selection should be shown simultaneously on both views. As the Braille display is limited (usually 40 Braille cells) most complex expressions will not fit on it, so the location of the Braille display content should appear clearly on the screen. We should remark that this is in the context of a teaching situation, so privacy issues are not essential here. However in the resulting software we plan to allow the user to hide the visual display, in the case of an exam for instance.

Synchronisation applies also to the caret position, and this allows pointing at a specific location. This will make it possible for the pupil to show a location using cursor routing buttons (and then make it appear clearly on the screen), and conversely clicking with the mouse on the graphically displayed expression should make the corresponding location appear clearly on the screen.

Let us add here that Mathematical expressions are embedded in text. The synchronisation and localisation of the current content displayed on the Braille device obviously has to be operational for text too.

Navigation

The second main feature that we need to implement in MaWEn is called navigation. This refers to the possibility of presenting various views of the current mathematical expression, where some branches are collapsed into meaningful chunks. The user can move the focus in the mathematical expression and choose to collapse one or several branches in order to have an overview, or to expand a collapsed one, in order to access its content. The user should also be able to collapse the whole tree in full, that is collapse every sub-branch so the expression can be examined progressively by opening the different branches one after the other.

We also wish to stress that navigation will enable inspection of Braille representations based on a standard linear view of formulas, that is without specific grammar. Indeed while the aim of these specific codes is to reduce the length of formulas they also make them extremely difficult for pupils to learn. With a collapse/expand feature the total length of the formula is no longer a problem. This collapse/expand feature obviously has to be synchronised as well.

Support functions

Finally carrying out mathematical calculations is even more difficult than reading and writing formulas. The problems involved in doing formal manipulations arise because of the complex structures that unfold during a calculation: sighted people may organise a computation such that it can be easily looked over and navigated, and they use additional graphics to help their reasoning. There is then a need for powerful editing tools that pupils can use to do calculations with more ease. These tools should provide support related to the Braille notation itself and not to the mathematical content. Additionally there is a tremendous need for providing contextual support for Braille mathematical code [17]. Such features will be implemented in MaWEn and have been designed and evaluated in previous works.

Editing

In editing mathematical expressions, we cannot yet rely on transcribers from Braille formats to MathML although various research projects aim at making these conversions possible. It was therefore decided to implement a hybrid entering mode, where simple terms could be entered directly from the keyboard (eventually from a Braille keyboard) while complex structures (fractions, roots, etc.) would be entered via commands (menus and keyboard shortcuts).

4.Universal Maths Conversion Library

One problem users have to deal with is that most mathematical software using Braille is associated with one specific mathematical Braille code. So for instance, if looking for a tool to convert LaTeX documents to Braille, one will find Labradoor [5], but this will be useful only if the reader knows the Marburg code.

To address this issue the UMCL (Universal Maths Conversion Library) was developed [2]. UMCL is a programming library encapsulating various converters for different Braille codes and which enables developers of applications designed for visually impaired people to easily support multiple Braille mathematical notations. UMCL is an open-source project.

The library can be used just as well with transcription tools (from mainstream notations to Braille and vice versa) as with software that needs real-time conversions (like formula browsers or editors). It also makes converting a document from one national Braille notation to another possible, increasing de facto the number of documents available for students and allowing blind mathematicians from different countries to exchange documents.

Overview

To make this possible without increasing the complexity an architecture based on a MathML (Presentation MathML) as the central representation of the formula was developed. The architecture of UMCL is quite simple. It includes a main module and as many input and output modules as there are Maths codes available. Input and output modules are independent and can be installed later on. The main module detects the available input and output modules and calls the modules needed to perform the conversions according to requests. In other words it is possible to add input or output modules to any application using UMCL, at any time a module becomes available.

Designers of client applications can use it easily through a simple and unique API. UMCL was developed in standard portable "C" with wrappers to various other programming languages. Currently output modules have been developed for the French notations (revisions 1971 and 2001), for the Italian notation and for the Marburg notation (German). A beta version for the British code is also already available. Input modules for LaTeX and Marburg Braille are also under development.

Canonical MathML

The central representation mentioned above is in fact a subset of MathML which we call "Canonical MathML" [3]. This Canonical MathML is an attempt to unify MathML structures in a deterministic way so as to simplify transcription into Braille. Canonical MathML was developed in order to make the development of output modules easier and to speed-up the processing time of Mathematical expressions. It must be stressed that Canonical MathML is valid MathML so it can be used with common tools which handle MathML. But the advantage of Canonical MathML is that all mathematical structures that are necessary to perform a correct transcription into mathematical Braille must be coded in a unique way. When processing a MathML formula for conversion into Braille about 90% of the processing time is needed for the canonisation.

It was initially decided to use Presentation MathML for two main reasons. The first reason was that it is a standard and there are numerous tools available to create, process or display MathML. Most mathematical applications dedicated to visually impaired people are based on or support MathML [6]. One of the aims of the library is to use it for designing transcription software, allowing professional transcribers to create documents with standard word processing software and to convert them into Braille. Nowadays most word processing software supports MathML. Another aim, as with MaWEn for instance, is to use it with reading/editing software. It is very easy to render MathML graphically within the Mozilla Application Framework, and other programming tools also do this.

Other representations of mathematical expressions based on semantics exists: for example Content MathML and OpenMaths. It is relatively easy to convert expressions from these formats to MathML presentation but the opposite is very difficult and some ambiguities cannot be automatically resolved. This leads us to the second reason. These ambiguities actually exist in the same way in any Braille Mathematical notation. For instance the expression "$a(x+1)$" could just as well be a function "a" applied to the expression "$x+1$" as the product of a variable "a" and the same expression. As in the mainstream case, the Braille code will be the same in both cases.

To produce Canonical MathML, all structures that need to be detected for conversion to Braille are looked for and for this reason we could say that it is partially a semantic notation, but the same ambiguities that are present in Braille are retained.

A UMCL input module for MathML is available and allows conversion of any MathML document to Canonical MathML.

UMCL Braille characters encoding

The Braille characters are depicted according to a specific Braille dots representation on 2 bytes. The first byte corresponds to a binary coding of the first column (dots 1-2-3-7) and the second byte to the second column (dots 4-5-6-8).

Two remarks should be made here. Firstly even if most mathematical Braille notation uses 6-dot Braille it seemed advisable to allow for support of 8-dot Braille with future applications. Secondly, although it would be possible to use a binary representation on only 1 byte, the maximum being exactly 8 dots, we need to be able to use it in XSLT stylesheets and XML documents. This would make using the 32 first characters of ASCII code (non printable characters) problematic.

Additionally it is a human readable code. For instance the letter 'a' in Braille is the dot 1 on its own: ⠁ and is coded '10' and the digit '7' in the French Antoine notation dots 1-2-4-5-6: ⠻, is coded '37".

5.MATHEMATICAL MODEL

Supporting synchronisation

The MaWEn Mathematical model is based on the Canonical MathML mentioned above. Synchronisation is obtained by setting IDs to each node of the MathML DOM.

All mathematical expressions are stored and processed using this format. This means that the working expression is always in this format whether or not the user is using Braille. Canonical MathML expressions are processed into Braille in real time by UMCL. The current MathML DOM is actually processed each time the view is modified (by navigation or editing). As input is always canonical the processing time by UMCL is very fast. The UMCL transcriptions carry the IDs so the Braille string is enriched. The actual output format of UMCL is a string containing a list of paired Braille symbols-ID separated by asterisks ('*'). The Braille symbol-ID pairs are separated by a dash and the ID may be empty: some Braille cells do not in fact correspond to one MathML tag.

The client application can then process this output, using selective information depending on its needs.

- Asterisks separate the different Braille symbols. As a Mathematical symbol cannot be split on 2 different lines, the division can only occur at an asterisk.
- IDs are used for easy retrieval of the corresponding Braille symbol or MathML tag when synchronisation is needed.

For example in MaWEn this allows the point/show feature in both directions: when a Braille cell receives an event (for instance, the cursor routing button attached to this cell is clicked) the MathML tag associated with the Braille symbol displayed on this cell is retrieved and the corresponding Mathematical symbols are highlighted on the graphical view. For the reverse action, if a MathML node receives an event (for instance when a Mathematical symbol on the graphical view receives a mouse click), then the corresponding Braille symbol is retrieved and underlined on the Braille display (using Braille dots 7 and 8, see below). This is also used to ensure that the selected area is the same on the two views.

Supporting Navigation

To support navigation we have introduced a '*mawen*' namespace. This namespace identifies attributes that we use to tag the MathML DOM.

The first attribute is '*mawen:fold*'. This attribute is boolean and indicates that the branch of which this node is the root is collapsed. The application calls on two MathML DOMs. One is the complete tree and one is the actual view (to be displayed on the screen). Each time a change occurs the actual view is re-processed from the original tree and each node containing the attribute 'mawen:fold' with the value 'true' is collapsed.

In some cases it is not the current node that should be collapsed but a parent one. This is the case for instance when an addition operation is enclosed between parentheses, e.g. 'a+(b+c)'. If the enclosed addition receives a collapse event the desired behaviour is to collapse the whole parenthesis group, in order to obtain 'a+Addition' instead of 'a+(Addition)'. In this case the ID of the parenthesis group is stored in the 'mawen:foldnode' attribute of the addition group.

When collapsed, each branch has a label. The label gives a clue to what the contents of this branch are. It may be a simple *'Term'*, a *'Fraction'*, a *'Square root'* etc. The labels are stored in the MathML tree using the '*mawen:type*' attribute. An additional abridged version of the label is also stored in the attribute '*mawen:shortname*'. This allows the application to offer the user an abridged view of the folded expression.

When a branch is collapsed its corresponding node, which contains a textual label or its abridged version given by the attributes mentioned above, receives the ID of the root node of the branch. There is no problem of duplicated IDs since the two versions cannot be in the same document at the same time.

As the navigation system is based on the MathML DOM it is also necessary to enrich this DOM with navigational information. Indeed a number of MathML nodes do not correspond to symbols either appearing on the graphical view or on the Braille display. For instance the '*⁢"* entity, which should be included in a <mo> tag, corresponds to an implicit multiplication and does not appear. This is also the case with most <mrow> tags. Tags that are not visible are marked using the attribute '*mawen:skip*' set to *true*. Additionally all nodes that are not skipped can be marked using '*mawen:next*' and '*mawen:previous*'. These attributes are set to the ID of the nodes which should be reached when moving to next/previous node. The ID of the entry point of the expression, i.e. the node corresponding to the first visible symbol of the expression, is stored in the '*mawen:firsttoken*' attribute of the root tag <math>. Similarly the last symbol is stored in '*mawen:lasttoken*'.

Mawen namespace attributes

Attributes to support collapse/expand feature	
mawen:fold (bool)	state of the node (collased or not)
foldnode (refId)	references the node which should be collapsed instead of the current one
mawen:type (string)	label representing the node when collpased
mawen:shortname (string)	abridged version of label
Attributes to improve navigation	
mawen:skip (bool)	node should be skipped
mawen:next (refId)	id of the next node
mawen:previous (refId)	id of the previous node
mawen:firsttoken (refId)	id of the first node (entry point)
mawen:lasttoken (refId)	id of the last node

Implementation

The first step when loading a formula is to be sure it is a Canonical formula, and that IDs are set. This must be ensured by the application using the model. The application must therefore canonise each expression and set IDs. This is carried out by calling the UMCL with the Canonical output. Canonisation will set the IDs. As each ID must be unique in a document, and a document may contain many expressions, IDs take the form 'prefix#eq:#sb', where #eq is the number of the formula in the document and #sb is the number of the MathML tag in the MathML expression. The prefix can be given as a parameter, but there is a default value ('formula'). The actual form of the ID is not important for the model to work, as long as each one is unique.

The MaWEN mathematical model is implemented as a class, called mawen.MathMLDOM which encapsulates the MathML DOM. It contains three main objects:
- the DOM source, the current complete tree of the formula, where all mawen:* attributes are set;
- the DOM view, the tree that must be displayed both in the graphical view and in the Braille view;
- a table of Braille symbol objects, which are instances of the BrailleSymbol class, containing attributes to store the Braille symbols (string of Braille characters) and a reference to the corresponding node in the MathML DOM.

A mawen.MathMLDOM object is instantiated with a MathML DOM. During initialisation, the DOM is processed and the mawen:* attributes are set. Then the DOM view is created by applying a XSLT stylesheet to the DOM source, and the table of Braille symbols is created by calling UMCL and processing its output: the UMCL output string is split according to asterisks, and for each paired Braille symbol-ID a BrailleSymbol instance is created. The Braille characters are converted from the UMCL Braille characters encoding to a specific Braille table (depending on user preferences). Then the MathML node corresponding to the ID is retrieved and its reference is stored in the BrailleSymbol instance. Each time the source tree is modified (either by collapsing/expanding a branch or by editing a node) the two latter steps are performed again: processing of the DOM view and conversion to Braille.

The XSLT stylesheet used for processing the DOM tree replaces the branch corresponding to each folded node by a <mtext> tag with the mawen:type attribute of the branch root node as textual content, and its ID as ID attribute. The UMCL is able to process these special nodes and retains their IDs.

6.IMPLEMENTATION IN MAWEN

Several MaWEn prototypes have been implemented to develop this model and to evaluate some of the desired features.

The MaWEn application is based on the Mozilla Application Framework. The user interface is implemented in XUL and Javascript, using CSS2 for look and feel. Then the different modules were developed in Javascript or C++, using XSLT stylesheets when necessary. External libraries are accessed via XPCOM objects implemented in C++. This is the case for instance with the UMCL, for which a management XPCOM object was implemented in Javascript and an interface XPCOM object was developed in C++.

The Mozilla Application Framework enables manipulation of DOM objects and also graphical rendering of MathML objects in XUL browser elements. The Braille output is managed by the libbraille library [14], which is also interfaced through an XPCOM object. Additionally a Javascript library was developed in order to show a virtual Braille display. It visually displays the Braille characters on the screen. This makes it easier for sighted users to see what is actually displayed on the Braille terminal. It is also very useful for working on implementation and doing demonstrations.

The work format of MaWEn is XHTML+MATHML, that is documents valid to XHTML with the MathML extension[1]. Additionally all MathML content in the documents must be Canonical. Actually the application may:
- open a MaWEn document, that is an XHTML+MATHML document where MathML content is canonical.
- import a document, or more specifically import a document, convert it to XHTML+MathML if necessary and canonise MathML contents. Note: for now only XHTML+MATHML documents can be imported.

When a document is loaded its DOM is managed by a class which allows synchronisation of textual contents on the graphical and the Braille displays. In this class a table of formulas is maintained, containing references to mawen.MathMLDOM objects instantiated to the Mathematical expression found in the document.

When the reader returns to a paragraph containing a Mathematical expression, the corresponding object is found again from the ID of the <math> tag (note that formulas may appear either on a single paragraph or embedded in the text). These objects are kept because the processing of these expressions may take a few seconds (due to transcription intp Mathematical Braille notation). For now these objects are created each time a formula is found during navigation in the document, but in the future it will be possible to create them in a separate thread in order to speed up the actual reading process.

MaWEn fully implements the synchronisation and navigation features described in section 3. The graphical view of an expression is displayed on the screen and the current Mathematical symbol containing the focus is highlighted with a coloured background. At the same time the corresponding symbol is underlined on the Braille display using dots 7 and 8. We use dots 7 and 8 to underline Braille characters because Mathematical Braille notations usually use only 6-dot Braille. During editing, the location of the caret is represented by dots 7 and 8 blinking.

The Braille user can click on the cursor routing key of a Braille character in order to show the corresponding Mathematical symbol on the screen. This symbol will then appear with a coloured background. In the same way the user can click on the formula to make any symbol appear underlined on the Braille display. The virtual Braille display also supports clicking and show the selection; additionally the underlined Braille characters appear with a coloured background.

Evaluation and further work

This work environment is currently in the process of evaluation in teaching situations. Evaluations are taking place in selected schools in France and in Austria. Some of these schools work only with visually impaired users while others are inclusive (for each class concerned only one pupil uses Braille). The teacher is asked to provide in advance the actual documents he is planning to use in the classroom on the date fixed for the evaluation. This document is prepared in the desired format. During evaluations the teacher is asked to present his lesson and exercises normally while Braille users use a prototype of the MaWEn software. After the evaluation sessions, both teachers and pupils are interviewed.

The next step will be to integrate the support functions and on-line help, described in section 3, in this environment.

[1]We use the following DTD:
"-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd

7.ACKNOWLEDGMENTS

The MaWEn prototypes are currently developed within the framework of the MICOLE project, which is funded by the INFSO DG of the European commission[2], within the framework of the IST 2002 programme (FP6/IST/eInclusion) under the reference IST-2003-511592 STP.

This paper was written with the support of the @Science project which co-funded by the INFSO DG of the European Commission, within the eContentPlus programme (FP6/thematic network priority), under the reference ECP-2005-CULT-038137.

8.REFERENCES

[1] Annamalai, A., Gopal, D., Gupta, G., Guo, H. and Karshmer A. INSIGHT: a comprehensive system for converting Braille based mathematical documents to Latex. *Proceedings of the Conference Universal Access in HCI - Inclusive Design in the Information Society (UAHCI 03)* (Heraklion, Greece, June 22-27, 2003). Lea, Mahwah, NJ, USA, 2003, 1245-1249.

[2] Archambault, D., Fitzpatrick, D., Gupta, G., Karshmer, A., Miesenberger, K. and Pontelli, E. Towards a universal maths conversion library. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2004)* (Paris, France, July 7-9, 2004). LNCS 3118, Springer, Berlin, 2004, 664-669.

[3] Archambault, D. and Moço, V. Canonical MathML to simplify conversion of MathML to Braille mathematical notations. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2006)* (Linz, Austria, July 12-14, 2006). LNCS 4061, Springer, Berlin, 2006, 1191-1198.

[4] Archambault, D., Stöger, B., Fitzpatrick, D. and Miesenberger, K. Access to Scientific Content by Visually Impaired People. *Upgrade.* CEPIS (2007), *to be published*.

[5] Batusic, M., Miesenberger, K. and Stöger, B. Labradoor, a contribution to making mathematics accessible for the blind. *Proceedings of the International Conference on Computer Helping People with special needs (ICCHP 98)* (Vienna, Budapest, August 31-September 4, 1998). OCG, Vienna, Austria, 1998, 307-315.

[6] Bernareggi, C. and Archambault, D. Mathematics on the web: emerging opportunities for visually impaired people. *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A 2007)* (Banff, Canada, May 7-8, 2007). IW3C2, 2007, proceedings on CD-ROM, 4 pages.

[7] Crombie, D., Lenoir, R., McKenzie, N., and Barker, A. math2braille: Opening access to Mathematics. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2004)* (Paris, France, July 7-9, 2004). LNCS 3118, Springer, Berlin, 2004, 670-677.

[8] Edwards, A., McCartney, H. and Fogarolo, F. Lambda: a multimodal approach to making mathematics accessible to blind students. *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2006)* (Portland, Oregon, USA, October 23-25, 2006). ACM, 2006, 48-54.

[9] Fitzpatrick, D. Mathematics: How and What to Speak. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2006)* (Linz, Austria, July 12-14, 2006). LNCS 4061, Springer, Berlin 2006, 1199-1206.

[10] Karshmer, A., Gupta, G. and Gillan D. Architecting an auditory browser for navigating mathematical expressions. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2002)* (Linz, Austria, July 15-20, 2002). LNCS 2398, Springer, Berlin, 2002, 477-485.

[11] Karshmer, A., Gupta, G., Pontelli, E., Miesenberger, K., Ammalai, N., Gopal, D., Batusic, M., Stöger, B., Palmer, B. and Guo, H. UMA: a system for universal mathematics accessibility. *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2004)* (Atlanta, GA, USA, October 18-20, 2004). ACM, 2004, 55-62.

[12] Moço, V. and Archambault, D. Automatic Translator for Mathematical Braille. Lea. *Proceedings of the Conference Universal Access in HCI - Inclusive Design in the Information Society (UAHCI 03)* (Heraklion, Greece, June 22-27, 2003). Lea, Mahwah, NJ, USA, 2003, 1335-1339.

[13] Raman, T.V. *Audio Systems for Technical Reading*. Ph.D. thesis, Department of Computer Science, Cornell University, NY, USA, 1994.

[14] Sablé, S. and Archambault D. libbraille: a programming library for easy access to Braille displays. *Proceedings of the 8th European Conference for the Advancement of Assistive Technology in Europe (AAATE 2003)* (Dublin, Ireland, Sept 1-3, 2003) IOS Press, Amsterdam, The Netherlands. 2003, 435-440.

[15] Stanley, P. B. and Karshmer, A. I. Translating MathML into Nemeth Braille Code. *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP 2006)* (Linz, Austria, July 12-14, 2006). LNCS 4061, Springer, Berlin, 2006, 1175-1182.

[16] Stevens, R.D. *Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People*. Ph.D. thesis, University of York, UK, 1996.

[17] Stöger, B., Miesenberger, K., Batusic, M. and Archambault, D. Multimodal interaction with mathematical formulae: Access, communication and support for blind people in doing mathematics. *Proceedings of the Conference Universal Access in HCI (UAHCI 05)* (Las Vegas, Nevada, USA, July 25-27, 2005). Proceedings on CD-ROM, 2005, 10 pages.

[18] Schwebel, F. *BraMaNet logiciel de traduction des mathématiques en braille*. 2003. [http://handy.univ-lyon1.fr/projets/bramanet/].

[2] The contents of this paper is the sole responsibility of the authors and in no way represents the views of the European Commission or its services.

Chapitre 3

Transcription

Voici une sélection d'articles traitant des outils de transcription permettant de produire des contenus dans des modalités accessibles à partir de contenus "ordinaires". Les 2 premiers concernent UMCL – *Universal Maths Conversion Library*, une initiative que nous avons mené pour intégrer ensemble, dans une libraire logicielle, un maximum de convertisseur de formats mathématiques. UMCL est distribué en *Open Source* et permet aujourd'hui de convertir du MathML ou du LaTeX vers plusieurs codes mathématiques braille (français, britannique, italien, allemand Marburg, américain Nemeth). Cette initiative a été lancée lors d'une des premières réunion du groupe iGUMA – *International Group for Universal Access* – début 2004. Le second décrit le format intermédiaire que nous avons retenu. Il s'agit d'un sous ensemble de MathML et les documents à ce format sont tous valides au sens MathML, mais nous avons introduit des restrictions afin de simplifier le passage au Braille. Les 2 derniers articles concernent les documents texte, ouvrages, revues ou simples feuilles d'exercice. L'idée est d'utiliser un format central, le format DTBook, qui permet de représenter de façon structurée et en incluant des balises sémantiques, pour représenter les documents et les stocker. Nous avons développé des outils pour générer des documents accessibles aux utilisateurs, en braille, gros caractères ou xHTML accessible (3.3, ainsi que des outils pour générer des documents au format DTBook à partir d'une suite logicielle gratuite très répandue (3.4).

3.1 [ICCHP'2004a]

Titre	**Towards a Universal Maths Conversion Library**
Auteur(s)	Dominique Archambault, Dónal Fitzpatrick[a], Gopal Gupta[b], Arthur Karshmer[c], Klaus Miesenberger[d] et Enrico Pontelli[e]
Affiliations	[b]Dublin City University, Irlande [b]University of South Florida at Lakeland, États-Unis [c]University of Texas at Dallas, États-Unis [d]Johannes Kepler Universität Linz, Autriche [e]New Mexico State University, États-Unis
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), LNCS 3118, Springer, Berlin
Editeur(s)	Joachim Klaus, Klaus Miesenberger, Wolfgang Zagler et Dominique Burger
Lieu/Date	Paris, Juillet 2004
Pages	664–669

Cet article présente les fondations d'UMCL, telles qu'elles avaient été proposées par les membres du groupe iGUMA que nous avions l'honneur de coordonner, et qui sont tous coauteurs. Les principaux chercheurs travaillant dans ce domaine en Europe (Universités de Linz-JKU, de Dublin-DCU et UPMC-Paris 6), et des États-Unis (New Mexico State University, University of Texas et University de Floride du Sud). Par la suite nous avons été rejoints (lors de la conférence ICCHP 2004 où était donné ce papier) par l'Université de Kuyshu au Japon. Nous avons implémenté le module principal et la moitié des modules linguistiques. D'autres membres d'iGUMA ont travaillé sur les autres modules.

Voir mémoire section **3.3.1**, page 37

Towards a Universal Maths Conversion Library

Dominique Archambault[1], Donal Fitzpatrick[2], Gopal Gupta[3], Arthur Karshmer[4], Klaus Miesenberger[5], and Enrico Pontelli[6]

[1] Université Pierre et Marie Curie, Inserm U483/INOVA,
9 quai Saint Bernard, 75252 Paris cedex 5, France,

[2] Dublin City University, School of Computing, Dublin 9, Ireland

[3] University of Texas at Dallas, Department of Computer Science
Box 830688; MS EC31, Richardson, TX 75083-0688, USA

[4] University of South Florida at Lakeland,
3433 Winter Lake Road, Lakeland, Fl 33803, USA

[5] Johannes Kepler Universität Linz, "integriert studieren - integrated study" (i3s3),
Altenbergerstrasse 69, A-4040 Linz, Austria

[6] New Mexico State University, Department of Computer Science,
Las Cruces, NM 88003-0001, USA

WWW home page: `http://lklnd.usf.edu/iguma`

Abstract. The study of Mathematics and Sciences have always been a difficult problem for blind students especially because of the complexity of Braille mathematical notations. Various projects developed converters allowing people to translate a formula from mainstream notations (like LaTeX or MathML) to Braille notations and vice versa. Today a new generation of tools aims at facilitating the understanding of the formulas by blind users, and the communication between sighted and Braille users. The project of Universal Maths Conversion Library is born from the decision of 6 organisations both American and European to join their efforts in that field.

1 Braille and Mathematical formulas

The writing of Mathematical formulas has always been a particular difficulty for Blind pupils and students. The following very simple fraction is usually written with a visual layout (1) that allows a very quick understanding of its semantics. The same formula written in a linear way (2) needs a bit more thought to be understood. Additionally it is written using many more symbols (11 while the first one only necessitates 7).

$$\frac{x+1}{x-1} \tag{1}$$

$$(x+1)/(x-1) \tag{2}$$

Braille is a linear writing system and consequently only the second form is possible. The first "natural" way of writing formulas is then to use this form, using eventually specific Braille characters for specific symbols (like the square root for instance). The problem is that formulas can be very long (the example shown above is quite simple but in the case of complex formulas the number of symbols increases dramatically), and then they are quite difficult to handle by blind students.

In order to reduce the length of these formulas as far as possible several Braille specific notations for Mathematics have been developed in different countries during the last century: Nemeth is used in the United States, Marburg is used in German speaking countries, there are two notations in French (the historical one was deeply revised in 2001), a British notation is used in United Kingdom and in Ireland, ItalBra is used in Italy.

These notations use context sensitive grammars that allow the use of the same symbol strings with different meaning depending on the context. They allow a significant reduction of the length of formulas. In counterpart these notations are quite difficult to learn (and to teach) and it is a fact that they have particular difficulties in the field of Mathematics and more widely in all scientific disciplines. Currently only very profficient Braille users are able to do it, while average mainstream pupils succeed easily. Indeed they have to deal with 2 difficulties: the Mathematical content itself and the Math code which is at least as difficult as the content.

To further complicate things, while the mainstream (visual) math notation is identical in every language, the same is not true for Braille notations. Then a Braille formula written using the Nemeth notation (American notation) is not understandable by a German speaking reader, used to working with the Marburg notation, and so on. This problem is quite important since the number of Braille documents is very small compared to the number of ordinary Maths books.

There exists an alternative way using a "graphical" embosser [1] which enables production of a tactile representation of the formula that is close to the visual layout, using Braille characters only for basic elements (like digits for instance). This method only works for printouts and can hardly be used by students to write formulas.

2 Conversion of Mathematical Braille

In the last twenty years, various projects have been developed in the field of Mathematical Braille notations, mainly with the aim of facilitating the written communication between sighted persons and Braille readers in the field of Mathematics (The MATHS Project [2], the LAMBDA project [3]). Various converters have been developed that performs conversions between mainstream mathematical formats like LaTeX and MathML and Braille notations:

Labradoor [4] (LAtex to BRAille DOOR) converts a full LaTeX document including Mathematical formulas into Marburg Braille.

Insight [5] converts formulas from Nemeth Braille code to LaTeX.

Bramanet [6] converts formulas from MathML to French Braille.

MMBT [7] (Multi-Language Mathematical Braille Translator) is an opensource project[1] allowing transcriptions from and to LaTeX, MathML, French (historical and new notation), British and Italian Braille notations (MMBT is still under development, some of those conversions are not yet available).

These converters are used for different purposes. One is to facilitate the production of scientifical Braille documents. For instance Labradoor allows a teacher to use a LaTeX document that he/she has prepared for his/her students and to convert it into Braille for a student. In the same way Bramanet produces Braille formulas from documents designed using mainstream Mathematical tools that have a MathML output (like MathType™ for instance).

In the other way (from Braille notations to mainstream formats) they allow sighted teachers or peers to access to formulas written by blind students.

3 A new generation of tools

Nowadays, the great efforts made around the MathML language and the progress made by rendering programs (like Mozilla for instance) and the equation editing software (like MathType) allow us to develop very useful software that might help the blind users to deal with the intrinsic complexity of Mathematical Braille notations.

3.1 MAVIS

MAVIS [8] is a formula browser that allows to navigate in the formula using voice output. It has been designed to convey the structure of the mathematical expression as well as its contents. MAVIS also uses the hierarchical structure of the formula to group subsections of an equation into meaningful “chunks”, enabling the user to navigate in this structure.

Studies in psychology allow us to enhance the way voice synthesis reads mathematical formula [9].

3.2 The Vickie Project

The Vickie Project[2] [10] aims at facilitating the inclusion of visually impaired pupils and students in mainstream education.

[1] http://mmbt.sourceforge.net, http://sourceforge.net/projects/mmbt/

[2] The Vickie Project is funded by the European Commission, on the IST (Information Society Technologies) Programme 2001 (FP5/IST/Systems and Services for the Citizen/Persons with special needs), under the reference IST-2001-32678.

A unified user interface is developed to enable visually impaired pupils and students to access documents using specific devices, while a synchronised graphical display supplies sighted teachers with a view of the same document. This interface is an application built on the framework of Mozilla[3]. The interface actually presents 3 views of the same document:

- a graphical view of the document, including images and mathematical formulas. This view is called the teacher's view.
- an adapted view to partially sighted users, with adjustable prints and colours.
- a Braille view. The libbraille library [11] is used to write directly on Braille devices.

The mathematical formulas are stored in documents in the MathML standard, and then can be displayed easily in graphical way thanks to the MathML renderer of Mozilla. This renderer allows to setup the size of print and the colours, for the adapted view.

The formula is automatically converted into Braille in real-time thanks to the MMBT converter [7] (see above). This software will also convert formulas written by the pupil in Braille to MathML in order to be displayed for the teacher.

In the Vickie project all documents are stored on a server [12] using the dtbook DTD format[4]. The mathematical formulas are included in the dtbook documents in MathML, thanks to the module mechanism of the dtbook DTD. Then the MMBT is also used to convert documents from a specific Braille notation that is used in France for printing (BrailleStar). Numerous documents are stored in that format and can then be imported into the Vickie server.

4 Next challenges

The first step will be to allow collaborative work between visually impaired and sighted people, each working on his representation of the same formula. Indeed the current tools allow real time conversions of formulas, enabling a sighted teacher to actually see in his natural way the line written by a pupil but he cannot show a point in this formula, in order to enlight an idea or to explain an error.

Actually doing mathematical calculations is even more difficult than reading and writing formulas. The problems in doing formal manipulations arise from the complex structures that deploy during a calculation: whereas sighted people may organise a computation such that it can be easily surveyed and navigated, blind persons tend to get lost quickly within a formal structure. There is a need for performant editing tools that pupils can use to do calculations more easily.

There is also a tremendous need for tools providing contextual support on the Braille mathematical code and doing calculations. These tools should provide

[3] The Mozilla project: http://www.mozilla.org

[4] The dtbook Document Type Definition (DTD) is part of the NISO Z39.86-2002 standard, also known as DAISY 3.0.

support about the Braille notation itself and not on mathematical contents. The aim is to reduce the gap between blind pupils and their sighted peers induced by the complexity of mathematical Braille notations.

5 iGroupUMA

To address these challenges, 6 organisations having expertise in the field of Mathematical Braille decided to join their efforts, creating the "International Group for Universal Math Accessibility" (iGroupUMA).

The iGroupUMA members are: University of Texas at Dallas, Dublin City University, University of South Florida at Lakeland, Johannes Kepler University of Linz, New Mexico State University, and University Pierre et Marie Curie in Paris.

6 Universal Maths Conversion Library

One of the goals of this group is to produce a programming library encapsulating various converters for various Braille codes in a single library usable through a simple and unique API. This library will be useful as well for transcription tools (from mainstream notations to Braille and vice versa) as for software that need real-time conversions (like formula browsers or editors). It will also make it possible to convert a document from a Braille national notation to another, increasing de facto the number of documents available for students and allowing blind mathematicians from various countries to exchange documents.

To make it possible without growing the complexity, it is necessary to adopt an architecture based on a central representation of the formula, and to develop parsers to generate this central representation from the different input formats, and output generators for each output format [13–15].

The iGroupUMA has decided to use MathML as central language. Input and output independent modules will be designed for each format: LaTeX, Nemeth, Marburg, British notation, the 2 French notations, ItalBra, etc. Those modules can be developed independently by each group. Then they will be integrated in a Universal Maths Conversion Library, that will offer any conversion from and to any formats. The input and output modules API will be published in order to allow other developing teams to create new modules.

References

1. Gardner, J., al: Simultaneous braille tactile graphics and ink with tiger ink attachment and duxbury. In: Proc. CSUN'04 (19th Annual Conference "Technology and Persons with Disabilities"), Los Angeles, California, USA (2004)
2. Edwards, A.: The Maths Project (2001) [http://www.cs.york.ac.uk/maths/].
3. Nicotra, G.: The LAMBDA Project (2004) [http://www.lambdaproject.org].

4. Batusic, M., Miesenberger, K., Stoeger, B.: Labradoor, a contribution to making mathematics accessible for the blind. In Edwards, A., Arato, A., Zagler, W., eds.: Proc. ICCHP 98 (6^{th} International Conference on Computers Helping People with Special Needs), Oldenbourg, Wien, München (1998)
5. Annamalai, A., Gopal, D., Gupta, G., Guo, H., Karshmer, A.: INSIGHT: a comprehensive system for converting Braille based mathematical documents to Latex. In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1245 – 1249
6. Schwebel, F.: BraMaNet logiciel de traduction des mathmatiques en braille (2003) [http://handy.univ-lyon1.fr/projets/bramanet/].
7. Moço, V., Archambault, D.: Automatic translator for mathematical braille. In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1335 – 1339
8. Karshmer, A., Gupta, G., Geiger, S., Weaver, C.: Reading and writing mathematics: The mavis project. Behavior and Information Technology **18** (1998) 2 – 10
9. Karshmer, A., Gillan, D.: How well can we read equations to blind mathamatics students: some answers from psychology. In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1290 – 1294
10. Archambault, D., Burger, D.: The vickie project. In Miesenberger, K., Klaus, J., Zagler, W., eds.: Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs). Volume 2398 of LNCS., Linz, Austria, Springer (2002) 90 – 97
11. Sablé, S., Archambault, D.: libbraille: a programming library for easy access to Braille displays. In Craddock, G., McCormack, L., Reilly, R., Knops, H., eds.: Assistive Technology – Shaping the future, Proceedings of the AAATE'03 Conference, Dublin, Ireland, Amsterdam, The Netherlands, IOS Press (2003) 435 – 440
12. an Integrated Publishing Chain for Accessible Multimodal Documents, T.: Benoît guillon and jean-louis monteiro and cédric chécoury and dominique archambault and dominique burger. In Klaus, J., Miesenberger, K., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs). LNCS, Linz, Austria, Springer (2004)
13. Karshmer, A., Gupta, G., Miesenberger, K., Pontelli, E., Guo, H.F.: The development of a tool to enhance communications between blind and sighted mathematicians, students and teachers: A global translation appliance. In Stephanidis, C., ed.: Proceedings of the UAHCI'01 conference — Universal Access in Human-Computer Interaction (joint with 9th International Conference on Human-Computer Interaction) — Towards an Informatino Society for All, New-Orleans, Louisiana, USA, Lawrence Erlbaum Associates (2001) 929 – 932
14. Batusic, M., Miesenberger, K., Stoeger, B.: Parser for the marburg mathematical braille notation. In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1260 – 1264
15. Palmer, B., Pontelli, E.: Experiments in translating and navigating digital formats for Matematics (a progress report). In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1245 – 1249

3.2 [ICCHP'2006]

Titre	**Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations**
Auteur(s)	Dominique Archambault et Victor Moço[a]
Affiliations	[a]Université Pierre et Marie Curie-Paris 6
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of ICCHP 2006 (10th International Conference on Computers Helping People with Special Needs), LNCS 4061, Springer, Berlin
Editeur(s)	Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler et Arthur Karshmer
Lieu/Date	Linz, Autriche, Juillet 2006
Pages	1191–1198

Voir mémoire section **3.3.1**, page 37

Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations

Dominique Archambault and Victor Moço

Université Pierre et Marie Curie, Inserm U483/INOVA,
9 quai Saint Bernard, 75252 Paris cedex 5, France

Abstract. This paper describes the Canonical MathML, a tentative to unify MathML structures in a deterministic way in order to simplify transcription into Braille. All Mathematical structures that are necessary to perform a correct transcription into Mathematical Braille are recognised and rewritten in a unique way. Additionally Canonical MathML is valid MathML so it can be used with common tools which handle MathML. The Canonical MathML was successfully used to build several transcribers from MathML to Braille national codes.

1 Introduction

Writing and reading Mathematical formulas has always been a difficult task for Blind people. Indeed sighted people use a graphical representation, in two dimensions, in which the layout helps the reader to understand the meaning of a formula. People with visual impairment use linear modalities, like Braille or speech synthesis, and therefore don't have access to this powerful help.

One of the main problems they encounter is the length of formulas, which tends to increase dramatically with the complexity of the formulas represented. In order to reduce this problem, the inventors of specific Braille notations to represent mathematical formulas used various techniques, including contextual grammars [1, 2].

These notations are well suited to the use by Braille readers, but their production has always been a problem. The main reason is that there are various such notations, corresponding to various languages, and the various location where they were developed. Then when a document is transcribed and exists in a digital format, it's very difficult to exchange with readers from another language (even though they can read the plain text in the other language itself without difficult). The same difficulty occurs with powerful software tools which have been developed in various laboratories and companies during the last two decades to produce Mathematical Braille (Labradoor (LAtex to BRAille DOOR) [3] produces Marburg Braille, Insight [4] handles Nemeth Braille, Bramanet works with French: http://handy.univ-lyon1.fr/projets/bramanet). Additionally some other very interesting projects handle MathML formulas, like the Math Genie [5] and Infty [6]. These projects would benefit of an easy way to access to a Braille output.

Today more and more digital documents are available via the Web, or in digital libraries exists [7]. MathML, maintained by the W3C Consortium became very quickly the standard for representation of mathematical contents in digital documents. Then providing converters from MathML to Braille notations will make life much easier for blind scientists and science apprentices.

In order to propose to users from different mathematical Braille cultures access to more documents, and access to a great variety of software tools, the International Group for Universal Maths Access started to develop a Universal Maths Conversion library [8]. This library is intended to provide conversion functionalities to and from various Braille notations, as well as standard representations, like MathML. Actually MathML was chosen by the members of the group to be the central representation.

The problem with MathML is that it allows to represent the same mathematical contents in many ways and then the conversion into Braille is not that easy. Furthermore Braille notations, if they are quite different from each other, at a certain level, share also a lot of characteristics. That's why we started to develop a set of tools to unify the way each mathematical structure is to be represented in MathML in order to simplify transcription into Braille. Actually 90% of the transcription work is done generically on the MathML representation, and this work is the same for any Braille code. Then the last part of the work, the transcription into Braille itself, is to implement the specific grammatical rules of the target notation, and to turn every mathematical symbol into its Braille representation, using a dictionary.

The intermediate MathML representation is specified so each Mathematical structure can be represented in a unique way, so the transcription into Braille is deterministic. In other words, all MathML structures that would be represented in the same way in Braille are unified into a unique MathML representation. We call this representation Canonical MathML. One important point about our Canonical MathML is that any Canonical MathML formula must be valid MathML in all cases.

2 Canonical MathML Specification

The next section presents the Canonical MathML structure, in different cases. Description of other structures (like limits) can be found on the umcl-demo website (see conclusion).

2.1 Subscripts and Superscripts

Subscript (respectively superscripts) must be represented using the `msub` (respectively `msup`) element. The element `msub` (resp. `msup`) must have 2 child nodes. None of them can be empty.

Subscripts with superscript must be represented using the element `msubsup`. It must have 3 child nodes. None of them can be empty.

The following table shows the Canonical MathML code to represent structures from these 3 cases:

Subscript	Superscript	Subscript with superscript

<msub>
  child1
  child2
</msub>
	<msup>
	  child1
	  child3
	</msup>
		<msubsup>
		  child1
		  child2
		  child3
		</msubsup>

Where: – child1 is a term, – child2 is an indice and – child3 is an exponent. Any of them can be simple elements (e.g. <mi>a</mi>) or complex elements encapsulated in a <mrow> element (This applies to all terms called child*i* in the rest of the paper, except when specified).		

Examples:

a_i	a^{i+1}
<msub> <mi>a</mi> <mi>i</mi> </msub>	<msup> <mi>a</mi> <mrow> <mi>i</mi> <mo>+</mo> <mn>1</mn> </mrow> </msup>

2.2 Parenthesis Groups

We call "parenthesis group" a group of elements delimited by an opening tag and closing tag. Then absolute values and intervals are considered as parenthesis groups. In MathML, there are several ways to represent parenthesis groups. The specific tag <mfence> allows to specify the opening and closing tags. But this tag is barely used by the tools which produce MathML. Generally, the group delimited by parenthesis are not explicitly defined on the MathML notation.

For instance, the formula $(a + b)$ is often represented in MathML as in the left column of the following table. In the canonical notation, all the group must be included in a <mrow> tag, as in the right column. Then the content of the parenthesis group must be included in a <mrow> too. This allows to differentiate algebraic parenthesis from intervals (see in the second example the representation of the interval $[0, +\infty[$).

NB: in the next examples, both representations are valid according to the MathML specification. There are a lot of ways to represent them, but only one in Canonical MathML.

$(a+b)$	
Valid in MathML	Canonical MathML
<mo>(</mo> <mi>a</mi> <mo>+</mo> <mi>b</mi> <mo>)</mo>	<mrow> <mo>(</mo> <mrow> <mi>a</mi> <mo>+</mo> <mi>b</mi> </mrow> <mo>)</mo> </mrow>

$[0,+\infty[$	
Valid in MathML	Canonical MathML
<mo>[</mo> <mn>0</mn> <mo>,</mo> <mo>+</mo> <mi>∞</mi> <mo>[</mo>	<mrow> <mo>[</mo> <mn>0</mn> <mo>,</mo> <mrow> <mo>+</mo> <mi>∞</mi> </mrow> <mo>[</mo> </mrow>

This representation makes it easier to navigate in mathematical formulas, as well as to translate formulas into mathematical braille. It is necessary to understand that converting a formula into canonical MathML does not modify the visual representation of formulas.

In the canonical MathML, a parenthesis is always represented as follows :

```
<mrow>
   <mo>opening symbol</mo>
   list of child nodes
   <mo>closing symbol</mo>
</mrow>
```

The list of child nodes can be:

- empty (e.g. $f()$);
- a simple element or a complex element included in a `mrow` in the case of algebraic parenthesis;
- a list of simple or complex elements in the case of intervals, where separators are simple `mo` elements.

2.3 Fractions

Fractions must be represented using the element `mfrac`. It must have 2 child nodes, none of them can be empty. The first child node correspond to the numerator and the second to the denominator.

The following table shows the Canonical MathML code to represent a fraction:

Fraction	Where:
<mfrac> child1 child2 </mfrac>	– child1 is the numerator, – child2 is the denominator.

Examples:

$\frac{1}{x}$	$\frac{1}{x+1}$	$\frac{x-1}{x+1}$
<mfrac> <mn>1</mn> <mi>x</mi> </mfrac>	<mfrac> <mn>1</mn> <mrow> <mn>x</mn> <mo>+</mo> <mi>1</mi> </mrow> </mfrac>	<mfrac> <mrow> <mn>x</mn> <mo>-</mo> <mi>1</mi> </mrow> <mrow> <mn>x</mn> <mo>+</mo> <mi>1</mi> </mrow> </mfrac>

2.4 Square Root, Roots

Square roots must be represented using the element `msqrt`. It must have only 1 child node, which cannot be empty.

Roots must be represented using the element `mroot`. It must have 2 child nodes, which cannot be empty. The first child node is the radicand and the second one the index.

The following table shows the Canonical MathML code to represent a square roots and roots:

Square root	Root
<msqrt> child1 </msqrt>	<mroot> child1 child2 </mroot>
Where: – child1 is the radicant, – child2 is the index (warning: the index is the second one).	

2.5 Summations, Products and Integrals

Summations, Products and Integrals must be represented using a `<mrow>` with 2 child nodes. The first child node is representing the opening of the structure. It may be a single element or a `<mrow>` of 2 or 3 child nodes, depending if the initialisation and limit are present. In any case the symbol representing the summation or product symbol must be there (basically Σ, Π or $\int$). The second one is the content of the summation or product. If a factor `dx` is present it must be included in the second child.

The following table shows the Canonical MathML code to represent summations in 3 cases:

Whole structure	limit is not there	Minimum structure
<mrow> <msubsup> child1 child2 child3 <msubsup> child4 </mrow>	<mrow> <msub> child1 child2 <msub> child4 </mrow>	<mrow> child1 child4 </mrow>
Where: – `child1` contains the summation or product symbol (Σ, Π or $\int$...), it **must** be a simple element. – `child2` is the bottom element (initialisation), – `child3` is the top element (limit), – `child4` is the content of the summation or the product.		

Examples:

$\Pi_i a_i$	$\int_{x=0}^{\infty} f(x)$
<mrow> <msub> <mi>Π</mi> <mi>i</mi> </msub> <msub> <mi>a</mi> <mi>i</mi> </msub> </mrow>	<mrow> <msubsup> <mo>∫</mo> <mrow> <mi>x</mi> <mo>=</mo> <mn>0</mn> </mrow> <mn>∞</mn> </msubsup> <mrow> <mi>f</mi> <mo>⁢</mo> <mrow> <mo>(</mo> <mi>x</mi> <mo>)</mo> </mrow> </mrow> </mrow>

3 Production of Canonical MathML

We have developed a set of 8 XSLT style sheets which allow to produce canonical MathML from any MathML source.

– `01_prepare.xsl:` process `<mrow>` tags (remove the ones that are not necessary. Save the ones which delimit complex structures, Sums, etc...
– `02_group1.xsl and 03_group2.xsl:` process opening and closing tags (like in parenthesis groups). NB: Currently only structures with an opening tag **and** a closing tag are supported.
– `04_unary_op.xsl:` search for unary operators and insert them into `<mrow>` when necessary. Add "InvisibleTimes" when necessary.
– `05_structure.xsl:` define XML structure for various specific MathML subtrees (e.g. sums, products, integrals).
– `06_binary_op.xsl:` process binary operators. This style sheet must be called 3 times successively with three different levels of operators.
– `07_canonical_mathml.xsl:` remove some temporary XML tags used by the other stylesheets that are not valid in MathML to obtain valid MathML.
– `08_add_index.xsl:` insert IDs to all MathML tags.

4 Conclusion

To produce Braille from Canonical MathML, we need to develop another XSLT stylesheet to implement the grammatical rules of the target Braille code, and a dictionary of symbols. A set of software scripts were developed to make it easier the development of the dictionary, using OpenOffice.org spreadsheet.

Currently the Canonical MathML and the corresponding XSLT style sheet allow to process any formula in algebra up to the end of secondary school. Implementation of matrices and limits is ongoing but should be available at the camera ready deadline.

Output style sheets and dictionaries were developed for French (2 notations) and Italian Braille codes. Marburg and British are under development in collaboration with universities of Linz and Dublin (DCU).

Additionally a portable C library, UMCL (Universal Math Conversion Library), was developed in order to make it easier the use of the converters. UMCL is an open-source project. A demonstration website gives access to documentation about the Canonical MathML, to an online conversion tool based on UMCL and to the source code:

`http://inova.snv.jussieu.fr/umcl-demo`

References

1. Archambault, D., Batušić, M., Berger, F., Fitzpatrick, D., Miesenberger, K., Moço, V., Stöger, B.: The universal maths conversion library: an attempt to build an open software library to convert mathematical contents in various formats. In Stephanidis,

C., ed.: Proceedings of 3nd International Conference on Universal Access in Human-Computer Interaction (joint with HCI International 2005), Las Vegas, Nevada, USA (2005) 9 pages (proceedings on CD-Rom).
2. Stöger, B., Miesenberger, K., Batušić, M.: Multimodal interaction with mathematical formulae: Access, communication and support for blind people in doing mathematics. In Stephanidis, C., ed.: Proceedings of 3nd International Conference on Universal Access in Human-Computer Interaction (joint with HCI International 2005), Las Vegas, Nevada, USA (2005) 8 pages (proceedings on CD-Rom).
3. Batusic, M., Miesenberger, K., Stoeger, B.: Labradoor, a contribution to making mathematics accessible for the blind. In Edwards, A., Arato, A., Zagler, W., eds.: Proc. ICCHP 98 (6^{th} International Conference on Computers Helping People with Special Needs), Oldenbourg, Wien, München (1998)
4. Annamalai, A., Gopal, D., Gupta, G., Guo, H., Karshmer, A.: INSIGHT: a comprehensive system for converting Braille based mathematical documents to Latex. In Stephanidis, C., ed.: Universal Access in HCI – Inclusive Design in the Information Society. Volume 4., Mahwah, New Jersey, USA, Lea (2003) 1245 – 1249
5. Gillan, D., Barraza, P., Karshmer, A., Pazuchanics, S.: Cognitive analysis of equation reading: Application to the development of the math genie. In Klaus, J., Miesenberger, K., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs). Volume 3118 of LNCS., Linz, Austria, Springer (2004) 630 – 637
6. Suzuki, M., Kanahori, T., Ohtake, N., Yamaguchi, K.: An integrated ocr software for mathematical documents and its output with accessibility. In Klaus, J., Miesenberger, K., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs). Volume 3118 of LNCS., Linz, Austria, Springer (2004) 648 – 655
7. Guillon, B., Monteiro, J.L., Chécoury, C., Archambault, D., Burger, D.: Towards an integrated publishing chain for accessible multimodal documents. In Klaus, J., Miesenberger, K., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs). Volume 3118 of LNCS., Linz, Austria, Springer (2004) 514–521
8. Archambault, D., Fitzpatrick, D., Gupta, G., Karshmer, A., Miesenberger, K., Pontelli, E.: Towards a universal maths conversion library. In Klaus, J., Miesenberger, K., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs). Volume 3118 of LNCS., Linz, Austria, Springer (2004) 664–669

3.3 [ICCHP'2004b]

Titre	**Towards an Integrated Publishing Chain for Accessible Multimodal Documents**
Auteur(s)	Benoît Guillon[(a)], Jean-Louis Monteiro[(a)], Cédric Chécoury[(b)], Dominique Archambault et Dominique Burger[(a)]
Affiliations	[(a)]Association BrailleNet [(b)]Université Pierre et Marie Curie-Paris 6
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs), LNCS 3118, Springer, Berlin
Editeur(s)	Joachim Klaus, Klaus Miesenberger, Wolfgang Zagler et Dominique Burger
Lieu/Date	Paris, Juillet 2004
Pages	514–521

Les outils de conversion décrit dans cet articles sont en production sur le serveur de la bibliothèque Hélène[5] de l'associatino BrailleNet, qui offre plus de 5.500 ouvrages numériques accessibles aux personnes déficientes visuelles. Ce serveur, sécurisé, permet aux utilisateurs déficients visuels enregistrés d'accéder à des documents libres de droit ou non, dans un cadre légal et avec les autorisations des éditeurs propriétaires des droits.

Voir mémoire section **3.3.2**, page 38

[5] http://serveur-helene.org

Towards an Integrated Publishing Chain for Accessible Multimodal Documents

Benoît GUILLON, Jean-Louis MONTEIRO, Cédric CHECOURY, Dominique ARCHAMBAULT, Dominique BURGER

Association BrailleNet, INSERM U483, Université Pierre et Marie Curie
9, quai Saint-Bernard 75252 Paris Cedex 5, France
{benoit.guillon, jean-louis.monteiro, cedric.checoury, Dominique.archambault, Dominique.burger}@snv.jussieu.fr

Abstract. Digital techniques have created new and stimulating conditions to provide visually impaired people with a better access to written information. The Helene Server project, started in 2000 by BrailleNet, aims at creating technical solutions to help transcribers to rationalize adapted documents production in France. The first action performed was to create a national repository of adapted documents available to transcription centers. In two years, this repository has gathered more than 1.500 books provided by publishers or transcribers. But the great variety of digital formats used by transcribers now raises the problem of resources normalization. This article presents a production chain of accessible documents based on the dtbook XML format.

Problem to be solved

Since 2000, BrailleNet has developed and maintained the Helene Server [1] (http://www.serveur-helene.org), a collaborative Web server to gather files to produce adapted documents for visually impaired people. This national project came as the continuation of a European TIDE project on Secure Document Delivery (SEDODEL [2]). It is currently part of the Vickie European funded project [3].
The Helene Server aims at providing transcription centers with a safe repository for original and adapted files of books under copyright. It offers the common functionalities of a digital library with additional features to manage publishers' authorizations and files versioning. To ensure the access to publishers' files, a secured delivery mechanism based on S/MIME and digital certificates has been set up.

After more than two years of existence, the Helene Server has reached some of its objectives:

- more than 1.500 books are stored, about ¾ of them being under copyright,
- BrailleNet has signed contracts with about 80 French publishers to obtain authorizations and source files,
- 40 transcription centers have been certified and collaborate to produce Braille and large print documents from source files provided by publishers.

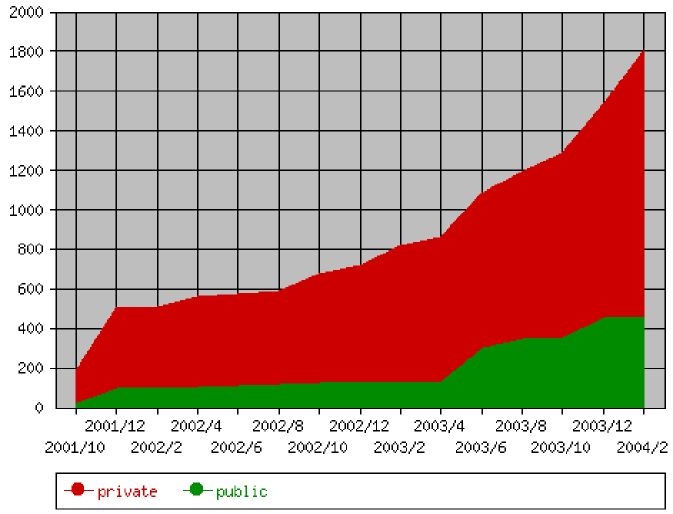


Fig. 1. Evolution of books number on the Helene Server since 2001

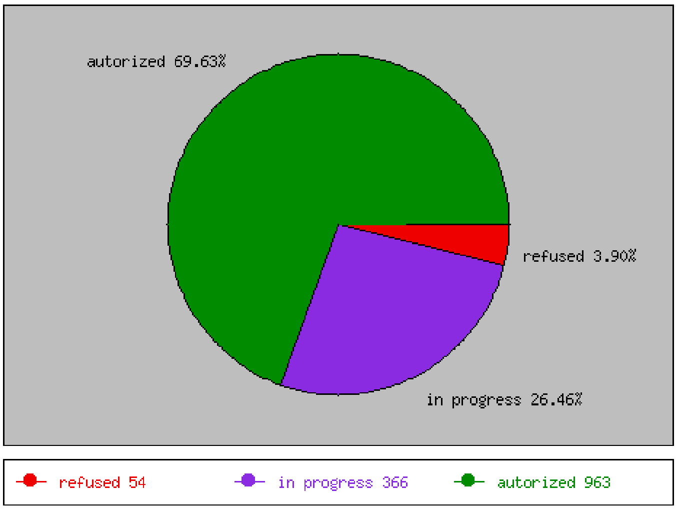


Fig. 2. Status of private books' authorizations

These results raise a new problem, related to the great variety of the formats used to store documents. Actually, more than 25 different formats of documents are registered, each of them used by transcription centers in different ways for different purposes.

This variety of formats raises questions from transcribers, for example:

- “This file was created using another operating system which uses a specific encoding for accentuated characters, how can I recode my file?”
- “This file was adapted for printed Braille using a specific layout format, how can I use it with another software? Can I use it for large print?”
- “I want to adapt this book which contains meaningful images, how can I insert descriptions of these images into the book file for the needs of blind people?”
- “Can I store a mathematical formula in a document so that it will be reused for Braille, large print or audio outputs?”

These few questions can be reformulated more technically:
- How to code structured documents that must be shared between different users in different environments?
- How flexible will this coding be as to be extended in the future?
- How to avoid adapting documents' layout for each output modality (Braille, large print, speech synthesis ...)?
- How to embed content's adaptation in a document so that they can be used selectively according to the modality used?

This article presents a complete solution to create documents in a unique high structured format and to convert them automatically into several specific accessible formats.

Technical principles

A central representation of documents based on dtbook

The dtbook [3] Document Type Definition (DTD) is part of the NISO Z39.86-2002 standard [4], also known as DAISY 3.0.
We have chosen dtbook as the central representation of documents because:
- it has emerged as a *de facto* XML standard to mark up general documents,
- it improves accessibility to people with print disabilities,
- it facilitates the output of documents in a variety of accessible formats,
- it provides the necessary elements to create semantically rich documents,
- it can be extended with external DTDs, to support specific document types (e.g. MathML for formula, poetry or drama modules ...),
- it can be processed using many existing tools on many platforms.

Collaborative enrichment rather than adaptation

Adapting a document aims at facilitating its access in any modality. But the adaptation process can not be fully automatic. It requires human expertise. Most of

the time this expertise is not concentrated in one center or person. For that reason the document adaptation process has to be collaborative, e.g. flexible and progressive.
In addition, the enrichment of the original document shall be independent of the targeted modality. The final presentation of the document will be parameterized later on when the document will be generated.
To achieve this, some constraints have to be respected in the document coding up stream. For instance:

- The general structure of the document should be marked up as to provide an easy navigation through it. Parts, levels, chapters (...) should be identified.
- The visual pagination should be preserved to allow visually impaired users to share references with sighted users.
- It should be possible to add conditional contents during the adaptation process. These additions may be specific to a modality, like alternative textual description of images, or alternative audio comments …).
- The document shall provide the necessary semantic information for its understanding using any modality (e.g. tables of data should be properly marked up for navigation; in a drama, speakers and speech should be distinguished ...).
- Special contents like mathematical formulas should be coded using a rich description format, as MathML [6].

Automatically generated output

The documents, if they were correctly coded and adapted by experts will contain the necessary information to generate automatically:

- Accessible HTML documents following the WAI [7] guidelines about accessibility, including document's logical structure (h1 to h6 elements), short alternative to images (alt attributes) and long descriptions when necessary, appropriate data tables structure for Braille and speech synthesis (thead, tbody, th elements and headers attributes).
- Files ready for Braille embossing, with a double pagination (Braille and original page numbering), navigation features (tables of contents, specific signalization of titles, textual descriptions of illustrations ...),
- a basic large print document including a double pagination and navigation features,
- files used as sources to record narrated books using the DAISY format (human voice or speech synthesis).

Technical implementation

To implement the solution proposed above, we have defined a development centered on the dtbook DTD. The objective is to progressively integrate tools into the Helene Server in order to obtain a consistent publishing chain to produce accessible documents. We will consider in this paper three types of tools:

Normalization tools

A first step in the chain is to rough out the main structure of documents. As the files provided by publishers and transcribers use a great variety of formats, a practical solution is to use a standard word processor, like Microsoft Word, for which most software programs offer exportation filters. Also MS Word is a word processor commonly used by transcribers. We developed a set of Visual Basic macros for MS Word for producing well structured files in RTF format.

These macros make easy:

- To create styles for the main structure elements of dtbook (headings, acknowledgments, bibliography, annex, lists ...)
- To mark-up original page numbers or to convert existing page break marks into page numbers,
- To import files prepared for Braille print into structured RTF files (convert specific ASCII characters into Word styles),
- To convert mathematical formula prepared with MathType [8] into MathML,
- To find bibliographical information on books on the Helene Server, and to fill in automatically the corresponding document metadata,
- To post the enriched document on the Helene Server in RTF.

In addition, the MS Word macro provides transcribers with a set of shortcut keys to markup documents faster.

Once uploaded on the Helene Server, RTF documents are converted into XML following the dtbook DTD using upCast [9] and appropriate XSL stylesheets.

The MS Word Macro can be found at
http://www.serveur-helene.org/braillenet/helene2/tools/macro/
The XSL stylesheets to convert upCast XML documents into XML dtbook documents can be found at
http://dsidtb.sourceforge.net/

A document enrichment tool

As MS Word does not provide the required functions to create the document structure described above, a more specific tool is necessary to allow transcribers to edit and enrich the structure and the content of dtbook files. This tool may be used on documents obtained with the MS Word macro or on documents outputted by automatic conversions from publishers' XML documents.

This tool should guide transcribers in the adaptation process, by providing them with simplified procedures to enrich documents. It should implement every features of the dtbook document type and hide its complexity to transcribers.
The document enrichment tool functionalities are:

- To apply particular styles to a text selection (Braille specific styles as “do not contract this region”, or theatre specific styles as “speaker”, “speech” …)
- To mark a text selection as being in a particular language,
- To add production notes to the document and to specify a target modality when necessary,
- To add production notes, short alternatives and captions to images included in the document,
- To markup data tables so that they will be easily understood by blind readers using speech synthesis or Braille displays,
- To create and modify the original pagination of the document and to create more complex numbering schemes.

This tool is currently developed within the Vickie Project.

In addition, we developed a tool to monitor dtbook documents adaptation level. This tool automatically generates a report on the accessibility features contained in a dtbook document. This report includes:

- the table of content of the document,
- the original page numbers,
- the number of images and if they have short alternatives, production notes and captions,
- the number of tables and if they are accessible to blind readers, if they contain a caption and a summary,
- the number of MathML formulas the document contains.

This tool is integrated to the Helene Server and is useful for transcribers to check the quality of dtbook documents before using them.

Output conversion tools

The enriched dtbook files can be used for producing:

- accessible HTML documents using customized navigation features (several HTML files linked one to each other using table of contents, navigation links on top and bottom of each file to reach previous or next section), link to particular CSS stylesheets and most of accessibility features recommended by WAI…
- Duxbury files (dollars-code format). Resulting files can imported directly into DBT and then embossed in Braille.
- PDF for large print which may be more specifically adapted to final user’s disabilities later on (for instance images size and colors …)
- Text only Daisy 3.0 package: an ncx file for navigation and a smil files for synchronization are automatically generated from a dtbook file. The dtbook file is modified to include references to smil files. Used with an opf file including book’s metadata generated by the Helene Server, a full text-only Daisy 3.0 package is built. This package can be read with a Daisy 3.0 player.

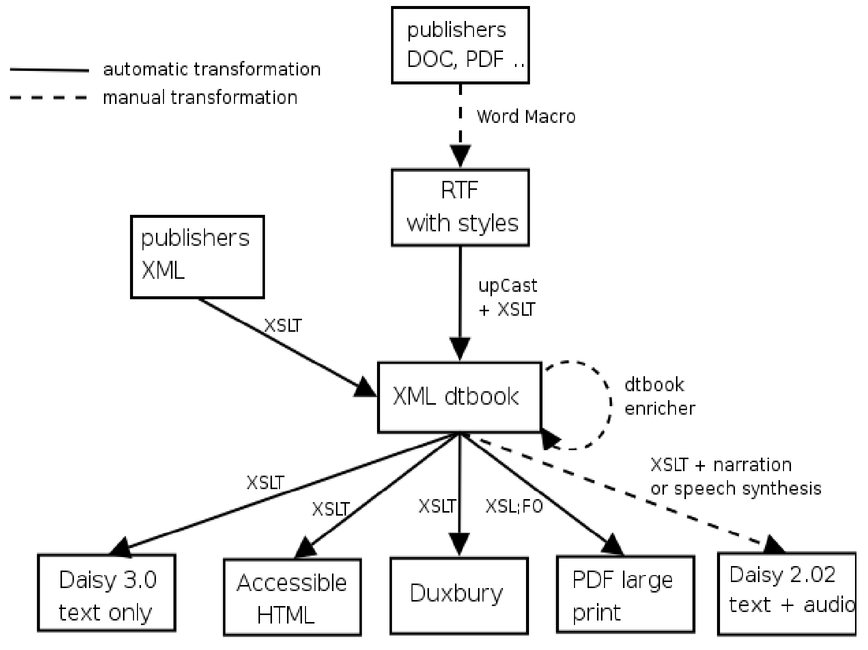


Fig. 3. The full production scheme

We developed XSL stylesheets to convert dtbook files in these different formats. These stylesheets are embedded on the Helene Server and are launched on the fly upon request of the user. The user can customize converters' behavior at generation time so that the output document will be more adapted to the final users needs (font size for large print, contraction or not for Braille, CSS stylesheets to be linked to HTML documents …). Mathematical formulas are translated into Braille notation using MMBT [10].

The stylesheets we developed can be found at http://dsidtb.sourceforge.net/

More specific software may also use these files to produce synthetic audio Daisy 2.02 files, like the software produce by Phoneticom [11].

Conclusion

The production scheme we exposed provides transcribers with a complete environment to create accessible documents whatever the target modality.
It avoids performing several adaptation works on a same book and helps transcribers to focus more on complex tasks as providing textual alternatives to images for Braille or images adaptation for large print.
As the conversion tools are fully integrated to the Helene Server, generating adapted documents is simplified and can be customized to fit the needs of final users.

Acknowledgment

The Vickie project is funded by the European Commission[1], on the IST (Information Society Technologies) Programme 2001 (FP5-IST-Systems and Services for Citizen/Persons with special needs), under the reference IST-2001-32678.

References

[1] Guillaume du Bourguet, Benoît Guillon, Dominique Burger : Helene: a collaborative server to create and securely deliver documents for the blind, proceedings of AAATE 2003

[2] Dominique BURGER, Iain MILLNS, Emmanuel SCHWARZ, Tom WESLEY: SEDODEL: A solution to the copyright barrier to access to information, CSUN 1999

[3] Dominique Archambault and Dominique Burger: The Vickie Project In: Proceedings of the ICCHP 2002 (8th International Conference on Computers Helping People with Special Needs), Linz, Austria, July 2002

[4] Digital Talking Book Expanded Document Type Definition Documentation for Version V110
http://www.loc.gov/nls/z3986/v100/dtbook110doc.htm

[5] Specifications for the Digital Talking Book
http://www.niso.org/standards/resources/Z39-86-2002.html

[6] W3C MathML
http://www.w3.org/Math/

[7] W3C Web Accessibility Initiative
http://www.w3.org/WAI/

[8] Design Science MathType
http://www.dessci.com/en/products/mathtype/

[9] Infinity-loop UpCast engine
http://www.infinity-loop.de/

[10] MMBT : Mathematical Multilanguage Braille Translator
http://sourceforge.net/projects/mmbt/
Victor Moço, Dominique Archambault: A transcription tool for mathematical Braille proceedings of AAATE 2003, Dublin, Ireland, September 2003

[11] Phoneticom
http://www.phoneticom.com/

[1] The contents of this paper is the sole reponsability of the authors and in no way represents the views of the European Commission or its services

3.4 [AAATE'2009]

Titre	**odt2dtbook – OpenOffice.org Save-as-Daisy Extension**
Auteur(s)	Dominique Archambault et Vincent Spiewak[(a)]
Affiliations	[(a)]Université Pierre et Marie Curie-Paris 6
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of AAATE 2009 (Assistive Technology from Adapted Equipment to Inclusive Environments) Assistive Technology Research Series, Vol. 25
Editeur(s)	Pier-Luigi Emiliani, Laura Burzagli, Andrea Como, Francesco Gabbanini, Anna-Liisa Salminen
Lieu/Date	Florence, Août 2009
Pages	212–216

odt2dtbook a été développé en collaboration étroite avec le consortium Daisy[6]. Ce projet Open Source a obtenu un *Gold Award* du programme d'innovation de la communauté OpenOffice.org en septembre 2008. Par la suite le projet a été repris par l'Université de Louvain dans le cadre du projet ÆGIS, et étendu sous le nom de odt2daisy[7].

Voir mémoire section **3.3.3**, page 39

[6] `http://www.daisy.org`

[7] `http://odt2daisy.sourceforge.net`

odt2dtbook

OpenOffice.org Save-as-Daisy extension [1]

Dominique ARCHAMBAULT [2] and Vincent SPIEWAK
INOVA/UFR919 – Université Pierre et Marie Curie-Paris 6 – France

Abstract. Odt2book is an extension to OpenOffice.org 3.0, which enables users to export any kind of document that can be opened by OpenOffice.org Writer into a valid DTBook file. Part of the DAISY standard, DTBook became recently a standard format in itself for storing and exchanging specific structured documents for print impaired people. Additionally it can be used to create a full Daisy audio-book. In this paper we will introduce the DTBook format and explain why such tools are particularly important. Then odt2dtbook will described.

Keywords. Daisy, XML, word processor, open-source, DTBook, accessible documents

1. Introduction

odt2dtbook is an extension to OpenOffice.org 3.0 which allows to export a document to the DTBook format. DTBook is an XML format, part of the DAISY standard (Digital Accessible Information SYstem) for audio books, which is nowadays the world's most widely used format for Talking Books for print disabled users [4]. Practically, a Daisy audio-book is composed of a set of files including audio files (at the mp3 format), synchronisation files and a DTBook XML file containing the whole contents of the book, together with structure and semantic information. In other words, this DTBook file contains all the text of the book, with structural and semantic tags. It is often compared to the backbone of a Daisy book. Actually a single DTBook file contains all the necessary information to generate a full DAISY book if needed. Furthermore this format is now a standard in itself. It was adopted as an ANSI/NISO standard (reference Z39.86), and it is used, as a storage format for books, by a large number of Digital Library for the Blind around he world (for instance TPB – the Swedish Library for the Blind –, The Library of Congress in the US, BrailleNet in France, etc...). [3] shows out the essential roles that many libraries for the blind are playing in the standards and open source software.

There is another very important application of the DTBook format within the framework of the US IDEA (Individuals with Disability Education Act, 2004). The IDEA requires that public schools make available to all eligible children with disabilities a free and appropriate public education in the least restrictive environment appropriate to their individual needs. The format for provision of accessible alternate format versions of ped-

[1] http://odt2dtbook.sourceforge.net

[2] Corresponding Author: D. Archambault, INOVA/UFR919 UPMC-Paris 6, B166, 4 place Jussieu, 75252 Paris Cedex 5, France; E-mail: dominique.archambault@upmc.fr.

agogical documents, called NIMAS 1.1 (National Instructional Materials Accessibility Standard), is aligning with Daisy.

Therefore it is very important to have some powerful and easy-to-use software applications enabling users, as professional transcribers working for libraries for the blind, or mainstream teachers having to provide print impaired students with NIMAS documents, to produce a valid DTBook. [5] have showcased a number of open source tools dealing with the DAISY standard, as well reading and authoring tools, including odt2dtbook.

2. DTBook

The DAISY[3] consortium has specified the DTBook XML DTD and is maintaining it. The Daisy consortium is also driving or collaborating to projects involving DTBook, like software applications to produce, maintain or access DTBook documents. For instance, as stated previously, a DTBook file can be used to generate a full DAISY book, with a speech synthesiser. This is possible with the DAISY pipeline[4], which is developed by the DAISY consortium.

The big advantages of the DTBook format are:

- **Standard**: it is a standard format, and a lot of software tools exist already to generate, handle and access DTBook documents;
- **Specific:** it is a specific format for print disabled people, which makes it easier to negotiate with publishers. for instance it does not contain presentation information;
- **Rich**: It contains all semantic and structural information about the document which allows to generate output documents fitting the needs of the readers. For instance [2] has implemented a set of XSLT style-sheets in order to generate printouts for partially sighted people, where a large number of settings can be made, and Braille documents at several standards;
- **Flexible:** it can be used to store as well novels, where the structure is minimalistic, as well as complex documents containing highly structured information or even Mathematical contents (the MathML-in-Daisy extension allow to insert Mathematical content in the MathML format). Additionally it provides the possibility to include adaptation notes, dedicated to specific modalities, in the documents.

A DTBook has 2 elements: a header and a body. The header (tag **<head>**) contains information about the book and the body (tag **<body>**) the contents itself. The body may be split in 3 parts: **<frontmatter>** (e.g. title, author, book jacket material, foreword, acknowledgements, dedication, and table of contents), **<bodymatter>** (the basic content of the document) and **<rearmatter>** (appendices, bibliographies, indexes, etc.). Then the content of the book itself is structured by **<level>** elements. Each level corresponds to hierarchical elements of the documents, like parts, chapters, or sections. Each level may include a heading and a number of other levels of minor degree (like parts may include chapters). A class attribute may be set to specify the kind of hierarchical element (e.g. **<level1 class="chapter">** would be used in a document where chapters are the higher level elements).

[3] http://www.daisy.org
[4] http://www.daisy.org/projects/pipeline

DTBook allows also to keep the page numbering information. This may be useful if some references are given to pages, referring to the printed book pages. For instance a teacher in an inclusive classroom may ask pupils to open their book at page 12. In that case it is important for a print impaired pupil using a DTBook document to be able to find it.

3. odt2dtbook

Odt2dtbook is made of 2 parts. One is a library that performs the conversion from an ODT file (OpenDocument Text file). This library is developed in XSLT and Java and it is independent of OpenOffice.org libraries, which makes it possible to use it in a standalone software. For instance odt2dtbook can be easily integrated in the DAISY pipeline [5]. The other part is the extension to OpenOffice.org itself, which is built against the Java UNO interface, and uses the converting library. It is necessary to install it within OpenOffice.org version 3.0 or any more recent. Both parts are cross-platform and have been tested on various platforms: Windows, Mac Os X, Linux Ubuntu, OpenSolaris.

Installation is very simple : the installation program installs both the extension and the converting library. The OpenOffice.org extension management systems will check automatically for updates. Once installed a new menu item appears in the *file* menu : *Export as Daisy*, just below the menu item *Export as PDF*.

To use the extension, the user has to give some structure to the document using the ordinary styles of OpenOffice.org (*heading1*, *heading2*, etc) plus a set of specific styles that have been designed to cover specific features of DTBook. Odt2dtbook recognises a large set of tags (the complete list can be found on odt2dtbook website): abbreviation, acronym, address, computer code, prodnote, etc. The extension supports as well lists, notes, footnotes, endnotes, links, images. These elements are automatically converted from the original document to their DTBook equivalent. In most cases there is nothing special to do. For instance to insert a footnote, simply create the footnote using the standard *Insert/Footnote* menu item from OpenOffice.org. Only in certain cases some specific information is required. In the case of images, the user has to provide an alternative text, which must be filled in the *Alternate text* field of the OOo image option dialog box. Tables are supported provided that heading cells are be correctly identified. Front matter and rear matter can also be rendered, using specific section names (see odt2dtbook manual for technical how-to information).

A set of templates have been designed which allows to use more easily the specific styles for Daisy elements (as mentioned above). When creating a document, these templates are available via the menu *File/New/Templates and Documents*.

In case of Mathematical contents, the expressions must be inserted in the OpenOffice.org standard way, that is inserting a Mathematical object (via the menu item *Insert/Object/Formula*. In this way the Mathematical expression will be stored by OpenOffice.org as a MathML code, which will be inserted in the DTBook as recommended by the MathML-in-Daisy extension. Let us insist on the necessity of inserting mathematical contents as Formula Object and to avoid absolutely using images since these images are not accessible at all [1].

Page numbering can be included (or not) in the exported DTBook. If requested (by checking the *include page numbers* check-box in the export dialog – see next paragraph), all pages that have a footer or a header including a *Page Number Field* will receive a page number in the exported DTBook. Additionally, odt2dtbook allows to set manually the page numbering, for instance to render the original layout of a document which might need several pages for each original page (like a schoolbook).

When the document is correctly prepared, with structure styles, etc., we can start the export process. Let us mention here that this will make the document accessible not only to generate a DTBook file, but to generate any kind of file in an accessible way [5] : as for instance accessible HTML, accessible PDF or even accessible ODT file which can be accessed by blind users via OpenOffice.org (since it is really accessible for versions 3.0 and more recent. Note that OpenOffice.org accessibility on various platforms, using various access technology has been reported on the OpenOffice.org wiki[5]).

The export process starts by activating the *Export as Daisy* menu item. If no structure style is present, a dialog box will discourage the user to go further and encourage her/him to make it more accessible before going on. Then an *Export* box will request several information from the user : an UID, which is auto-completed by a generated one (this generated UID can be overridden if the user needs to have a UID depending on an organisation specific numbering, e.g. MyOrg-2009-05-001, for instance). The other fields are:

- Title (auto-completed from *File>Properties>Description>Title*)
- Creator (auto-completed from *Tools>Options>OpenOffice.org>User Data>First Name+Last Name*)
- Publisher (auto-completed from Creator)
- Producer (auto-completed from Creator)
- Language (auto-completed from the main language of the document, set in *Tools>Language>For all Text*). Note that if some paragraphs have different languages, a "lang" attribute will be set for the corresponding element.

Additional options allow to include page numbering (as mentioned above), to use the DTBook alternative level markup (useful for documents having more than 6 hierarchical levels), and to include a CSS file[6] (in order to allow the user to access the DTBook file from a simple Web browser).

Finally the exported DTBook file (with possible an attached CSS) is generated.

4. perspectives

Among other proposed ways to produce DAISY books, Microsoft is developing an OpenXML DAISY project but, for the moment, this does not support Maths and is not, obviously, built around a free open source platform. OOo with odt2dtbook is therefore the only rich, cross-platform, accessible, free and open source authoring environment for authoring DAISY books, which would be a tremendous addition to the disability community. It can be used for free by by anybody, in conjunction with OpenOffice.org, which is, as widely known, free and open source too. For instance it can be used by libraries for the blind who need to convert documents from any format supported by OOo (including ODT, Ms Word, etc.) to DTBook, by teachers who need to provide accessible documents,

[5]http://wiki.services.openoffice.org/wiki/Accessibility
[6]Cascading Style Sheet, http://www.w3.org/Style/CSS/

compliant with NIMAS, to disabled pupils or students and by resource centres who produce accessible documents in schools and universities. In addition, as OpenOffice.org has improved accessibility features, odt2dtbook makes it possible for print disabled people to author DTBook documents.

The development of the Daisy pipeline light by the Daisy consortium, which is a library allowing to created automatically a full DAISY book (using the system speech synthesis), makes it possible to generate directly within OpenOffice.org a full audio-book. This is currently under development and will soon be available, including in one of the next versions of odt2dtbook.

Acknowledgement

Odt2dtbook received in September 2008 a Gold Award from the OpenOffice.org Community Innovation Program (sponsored by SUN Microsystems).

http://odt2dtbook.sourceforge.net

References

[1] C. Bernareggi and D. Archambault. Mathematics on the web: emerging opportunities for visually impaired people. In *W4A '07: Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, pages 108–111, New York, NY, USA, 2007. ACM Press.

[2] G. du Bourguet, B. Guillon, and D. Burger. Helene: a collaborative server to create and securely deliver documents for the blind. In G. Craddock, L. McCormack, R. Reilly, and H. Knops, editors, *Assistive Technology – Shaping the Future, Proceedings of the AAATE'03 Conference, Dublin, Ireland*, pages 451–455, Amsterdam, The Netherlands, September 2003. IOS Press.

[3] G. Kerscher. The essential role of libraries serving persons who are blind and print disabled in the information age. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer, editors, *ICCHP*, volume 4061 of *Lecture Notes in Computer Science*, pages 100–105. Springer, 2006.

[4] G. Kersher. Structured access to documents, digital talking books, and beyond: The daisy consortium. In K. Miesenberger, J. Klaus, and W. Zagler, editors, *Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs)*, volume 2398 of *LNCS*, pages 1–2, Linz, Austria, July 2002. Springer.

[5] G. Kersher, M. Gylling, and D. Archambault. Open source software for daisy - today and in the future. In *Proc. CSUN'09 (24th Annual Conference "Technology and Persons with Disabilites"*, Los Angeles, California, USA, March 2009.

Chapitre 4

Interfaces spécifiques de jeux multimédia

Cette série d'articles concerne plusieurs interfaces spécifiques que nous avons réalisées afin d'étudier la façon de présenter différents types d'interaction de jeux de façon non visuelle. Le premier concerne la représentation sur un afficheur braille linéaire d'un espace de jeux à 2 dimensions, correspondant à un plateau de jeu simple de type “serpent” *(snake)* ou “labyrinthe” *(maze)*. Le suivant concerne un jeu multijoueur permettant à des utilisateurs handicapés visuels, moteurs ou à des valides de jouer ensemble. Le troisième article décrit une étude réalisée pour mettre au point un système d'auto-apprentissage non visuel, ne nécessitant pas d'instructions verbales. Le quatrième concerne le moteur de jeu spécifique que nous avons développée ans le cadre du projet TiM (projet financé par la Commission Européenne dans le cadre du programme IST). Enfin le dernier de cette série décrit un générateur de jeux audio/tactiles développé dans la continuité du projet TiM. Il permet de généraliser un des prototypes proposés durant le projet TiM, et fournit une interface graphique simple permettant de créer des contenus de jeux destinés à de jeunes enfants aveugles et malvoyants.

4.1 [GTDW'2005]

Titre	**Accessible video games for visually impaired children**
Auteur(s)	Alexis Sepchat[(a)], Nicolas Monmarché[(a)] et Dominique Archambault
Affiliations	[(a)]Polytech'Tours
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of GTDW 2005 (The 3rd Annual International Workshop in Computer Game Design and Technology)
Editeur(s)	Madjid Merabti, Newton Lee, Mark Overmars et Abdennour El Rhalibi
Lieu/Date	Liverpool, Grande-Bretagne, Novembre 2005
Pages	58–67

Voir mémoire section **3.1.3.5**, page 32

Accessible video games for visually impaired children

A. Sepchat
Université François Rabelais
Laboratoire d'Informatique
Tours, France
alexis.sepchat@univ-tours.fr

N. Monmarché
Université François Rabelais
Laboratoire d'Informatique
Tours, France
nicolas.monmarche@univ-tours.fr

D. Archambault
Université Pierre et Marie Curie
INSERM U483/Inova
dominique.archambault@upmc.fr

ABSTRACT
Currently, a few video games are accessible for visually impaired people. Our works deal with the development of such games and more precisely on tactile accessible video games for visually impaired children. The aim of these works is the establishment of a global method to develop such games. So, the first step is the conception of simple games as Snake and Maze ones. Then from these two games a general accessible video game scheme has been extracted. And finally, a semi automatic generator based on this scheme has been developed.

General Terms
Accessible video games

Keywords
Accessible video games, Visual impairment, Braille terminal

1. INTRODUCTION
Created in the first half of the 19th century by the French inventor Louis Braille, the Braille was an important designing issue for visually impaired people. Nevertheless, the volume of translated data is significant. It is why computer science has been a great revolution to develop Braille. It permits to associate a representation technique with a storage technique.

But currently, if computer science is a wonderful learning tool for visually impaired people, it keeps being especially a working tool and it does not have a play aspect as for other people without any visual deficiencies or with other kind of deficiencies.

In previous works in the bosom of the HaNT team of the University of Tours Computer Science Laboratory (L.I.), we have already focused on this play aspect and have developed a game generator for young children [3]. These works were a part of the TiM project (Tactile Interactive Multimedia) [2] which goal is the development of accessible video games for visually impaired children. Games developed were then based on sounds and textures.

In these new works we point our efforts on braille terminals, which are expensive devices used by older children than previously and adults. Braille learning is very difficult and very technical. So, it is very important to develop games based on Braille terminal to improve tactile sensibility playing and not only working.

On the other hand, people can not buy several Braille terminals, so games have to run whatever the Braille terminal used. It is why, we have used the "Libbraille" library [4] which offers several primitives of communication between the computer and the terminal disregarding the model.

Figure 1: ECO ONCE 20 Braille terminal

The first step of our work is based on the development of several tactile games as a Snake game and a Maze game. The difficulty is to find the best representation for the 2D space toward the Braille line of the terminal. Moreover, the size of the terminal, i.e. the number of Braille cells (a cell is represented in figure 2), is limited so the translated information has to be selected.

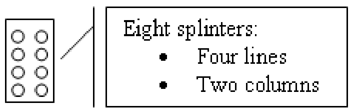


Figure 2: Braille cell detail.

The second step consists in the research of a mechanism to facilitate the development of such games. The extraction of a general structure from these two first games leads us to a semi automatic generator for accessible video games.

This paper is organized as follows: in section 2, we provide a short state of the art of accessible video games that has already been developed. In section 3 we give details about games that we propose. Section 4 give details about the game generator and section 5 points out improvements that could be done. Last, section 6 concludes this paper.

2. ACCESSIBLE VIDEO GAMES

2.1 Accessible video games in general

A large part of current video games can be played by impaired people [5] but many of these players can meet difficulties. These difficulties are linked with the speed of the game, its complexity, player perception problems, according to his/her disabilities.

Requirements for impaired players depend on the kind of handicap but also on the person himself/herself. Indeed, each person can live the handicap differently; the ability of the person depends on the level of the handicap which can evolue through time. So as it can be difficult to develop an accessible video game for one precise kind of handicap, it will be more difficult to conceive an all-disabilities accessible video game.

A way to solve this problem is to give large range of configuration with many parameters. But this range is very limited in current video games even if the demand exists. Impaired people, impaired children and their parents are still waiting for this kind of games.

Figure 3: Level Games - Aliens invasion.

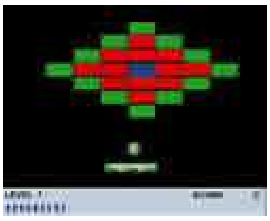


Figure 4: Level Games - Brick Out.

Nevertheless, some games can be found on the Web such as "level games" [6], a framework developed in 2000, which offers 3 accessible games (Alien invasion - figure 3 , Brick out - figure 4, Ruby Ridge - figure 5) or the "DOOM-like" game "Wheels" [7] (see figure 6).

These games can be parameterized according to child motive abilities: displacement and action speed, size of the racket or the aircraft, number of lives, etc. Moreover, these games can be used with several kinds of interfaces such as Intellikeys, trackball, joystick, vocal recognition, etc.

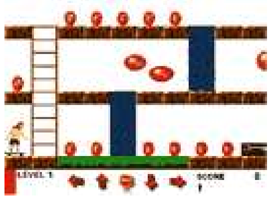


Figure 5: Level Games - Ruby Ridge.

Figure 6: Wheels.

2.2 Accessible video games and visually impaired people

Visually impairment is very restricting in video games because it prevents player from using the standard output device. So it is necessary to use sounds, vocal recognition or tactile output like a Braille terminal [5].

As in motive handicap, some works have been made to develop such games but there are only few ones. TiM European project focuses on this kind of games. On the same way, some games can be quoted: the car racing game "drive" has been developed by three students atUtrecht schools of the Arts; it is based on sounds and on the advices given by a virtual co-pilot; a "bomb disposal expert" (Swiss site "Blind Life") or a "naval battle" (ESSI students).

A website, JeuAccess[9], has been created, by Sabine Gorecki, to facilitate accessible video games access for visual impaired persons. It gathers several links for downloading this kind of games, documentations, informations, forum, adresses, ... and only deals with accessible video games (with the corresponding equipment) or with standard video games which can become accessible ones. It also offers a spreading list allowing members to inform others about new games, to organize tournaments, ...

Then, a very complete project is the video game "Tachido" [8] by Tsunami Factory (figure 7). Its aim is to create a video game shared by visually impaired persons and others. It is based on sounds and a force feedback lever. It is the story of a young samouraï who is losing his sight all along the adventure. So progressively, players without visually impairment have to play as visual impaired player. But this project keeps being developed.

But all these games are often based on sounds and vocal

Figure 7: Tachido game.

recognition. Even if Tachido offers a force feedback lever, there are very few tactile video games. So we focus more especially on this kind of accessible video games to help visually impaired people to improve their touch. So Braille learning can be improved by ply practise.

3. ACCESSIBLE TACTILE VIDEO GAMES

3.1 The two games developed

The two first games developed are the following:

- A maze game where the player moves inside a limited grid and has to find the exit, collecting objects and avoiding enemies.
- A snake game, as we can find on many mobiles phones, where the player moves inside a limited grid and have to eat apples which increase the length of the snake. The aim is to eat the maximum of apples without hitting the snake's tail or the gri borders.

These games are simple games based on the displacement of a character inside a grid.

3.2 Maze game, states and Braille representation

The Braille representation is the most important difficulty of this project. Even if, technically we are actually able to create many representations, it is difficult to understand blind people feelings, the way they represent 2D space and the way they read on the Braille terminal. This difficulty is increased by the fact that these games must be easy enough to be played by children.

3.2.1 The different states

Each cell of the grid can be in a given state and the number of different states is limited. In the maze game, the diffe are: character, empty, out of game, obstacle, enemy, object to pick up, exit. So each state is associated with a Braille representation i.e. a splinters configuration as for example (see figure 8):

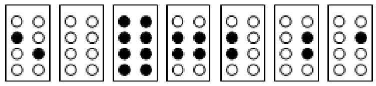


Figure 8: Maze game different states and splinters configuration associated (character, empty, out of game, obstacle, enemy, object to pick up, exit)

3.2.2 Simple representation

The Braille terminal has a limited length i.e. a limited number of cells. Consequently, the whole grid of the game can not be translated. It is necessary to select the most interesting game areas. So we have decided to translate the direct neighbourhood of the character as a window centered on the character. This window can take several patterns such 3×3 window, 1×20 (current character line) or 20×1 (current character column)(see figure 9):

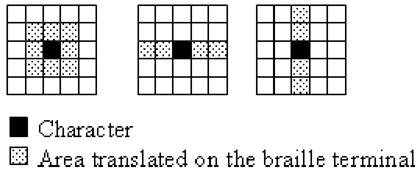


Figure 9: Maze game simple representation

Then, the representation on the Braille terminal consists in applying the window pattern chosen to the grid area. So, all the grid areas in this window will appear in the Braille line with the corresponding splinters configurations. The position of these areas in the Braille line is given according to the browsing strategy.

So, the construction of the Braille line is linked with the browsing strategy of this window. Indeed, this line is built in a sequential way so several constructions have been proposed:

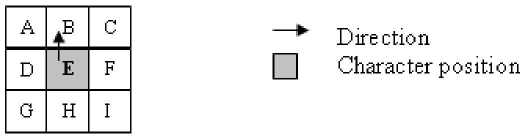


Figure 10: 3x3 window representation

- With a 3x3 window (see figure 10):
  - 3x3 window and line route (see figure 11):

A	B	C	D	E	F	G	H	I

Figure 11: Line route

  - 3x3 window and cyclic route taking care of the character direction (character position does not appear) (see figures 12 to 15):
- With a line or column window, several patterns are possible:

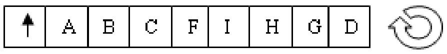


Figure 12: Cyclic route - North or no direction

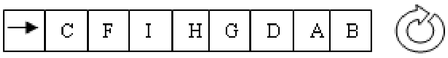


Figure 13: Cyclic route - East direction

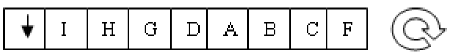


Figure 14: Cyclic route - South direction

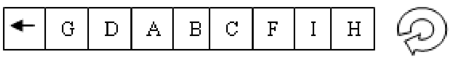


Figure 15: Cyclic route - West direction

- Current line of the character
- Current column of the character
- Current column or line of the character according to his direction (see figure 16):

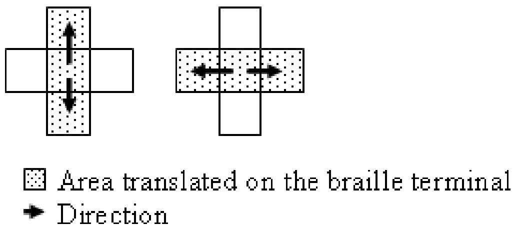


Figure 16: Line or column representation

In that case, it is easier to build the Braille line. Indeed, the pattern is already a 1D neighbourhood whereas the 3x3 window is a 2D one. It is why it is necessary to think about a way of browsing it in order to construct the corresponding 1D neighbourhood.

3.2.3 Direction and displacement

In these games based on the displacement of a character inside a grid, different kinds of displacement can be used. In our case, we manage two ones:

Multidirectional: the displacement is the same that the key pressed by the player (key up = up; key down = down; ...).

Forward, Backward and Pivot: the displacement depends on the current character direction. Keys up and down permit to move forward and backward according to the current direction. Whereas left and right keys permit to turn round itself so change the direction.

The way of displacement modifies the representation. In case of a forward, backward and pivot displacement, the displacement, depending on the current direction, so this information has to be translated in Braille representation. The first cell of the Braille line will be reserved to this representation (see figure 17):

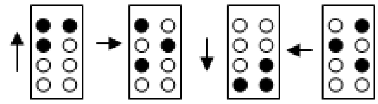


Figure 17: Direction representation (North - East - South - West)

Nevertheless, in the case of multidirectional displacement, direction is not useful. So direction will not appear in Braille line.

3.2.4 Enriched representation

As stated before, game cells can only take a limited number of states. For example, Maze game can take seven states (out of game, empty, character, exit, enemy, object, and obstacle). But each Braille cell gets 8 splinters allowing to represent $2^8 = 64$ states. So, it is possible to provide more information with one Braille cell than only one game area state. It is what we call enrichment.

This enrichment allows creating a larger window representating the indirect neighbourhood i.e. the neighbourhood further from the character than the translated window.

For example, in case of a 1×20 window, i.e. current character line representation, in Maze game (so seven states possible), the following representation can be used (see figure 18):

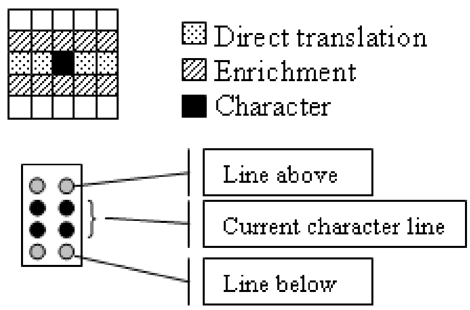


Figure 18: 1×20 window Maze game enrichment

The four splinters in the middle of the Braille cell represent the state of the considered game area. These four splinters allow to represent $2^4 = 16$ states. The configurations of splinters used are the following (see figure 19):

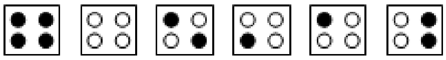


Figure 19: Direct neighbourhood and enriched representation (out of game or obstacle, empty, character, object, exit, enemy)

The four others splinters allow to represent the state of indirect neighbourhood which is composed by two elements: in that case, the line above and the line below the current character line. Two of the four splinters are associated with each element which allows making only $2^2 = 4$ configurations per element. Four configurations are not enough to translate the seven states of the game so it is necessary to cluster these seven states in four classes: empty, out of game or obstacle, interesting (exit or object to pick up) and danger (enemy). Each class is associated with a configuration of splinters (see figure 20).

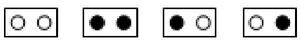


Figure 20: Indirect neighbourhood and enriched representation (empty, out of game or obstacle, interesting, danger)

A similar mechanism, using the notion of direct and indirect neighbourhood and the notion of class can be created for every pattern of window and way of browse.

3.3 Snake game, states and Braille representation

The second game developed is a Snake one. It is also based on a displacement inside a grid but some elements are different. First, the character is a list of coordinates which is getting larger and larger all along the game. Then, the displacement is made by a timer. The player just can choose the direction and the timer makes the character move forward. These two elements are the two main differences between the two games.

3.3.1 The different states

In this game, there are fewer states for each game cell. Indeed, there are only four states: empty, out of game, Snake (head or body) and apple. So each game cell can be represented on Braille terminal with fewer splinters, two are enough ($2^2 = 4$)(see figure 21).

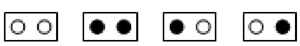


Figure 21: Direct neighbourhood (empty, out of game, Snake, apple)

3.3.2 Window pattern and representation

As we use only two splinters to represent a game cell, it is possible to represent a larger window. Indeed, it allows building a 4×20 window in our case. But four lines is not very useful: obviously we need the current line of the character, the line above and the line below. But which fourth line must be chosen? The second line above or the second line bellow? So it is better to represent only a 3x20 window and keep two splinters for the enrichment (see figure 22).

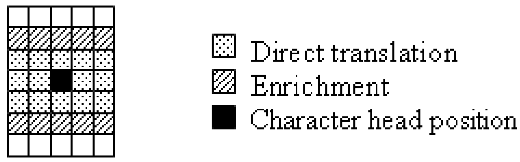


Figure 22: Direct neighbourhood in Snake game enriched representation

Then, two splinters are available to represent enrichment. The best thing to do is to give information about the neighbourhood above the window but also below it. So, one of the two splinters is associated with each of these neighbourhood (see figure 23).

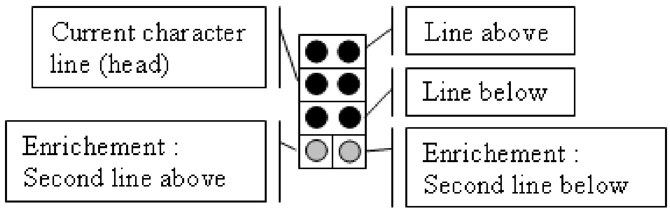


Figure 23: Snake game enrichment scheme

Nevertheless, one splinter is not enough to translate the four states so we cluster these states in two classes: danger (out of game or snake body) or interesting (empty or apple) represented by a splinter outside or inside.

Moreover, in the Snake game, the displacement is a multidirectional one. The player chooses a new direction without taking care of his current direction. So this information has not to be translated in the Braille line.

Finally, the development of these two accessible video games have permitted to underline the necessary mechanisms in order to create such accessible games.

These mechanisms deal with the structure of the applications, the way to represent the grid (or a limited area inside the grid) on the Braille terminal, ...

3.4 Presentation of applications

These two games are based on similar interfaces:

These interfaces are composed by three areas (see figure 24 and figure 25):

A : The first one is the game grid. It represents all the game cells with their associated state. This state is

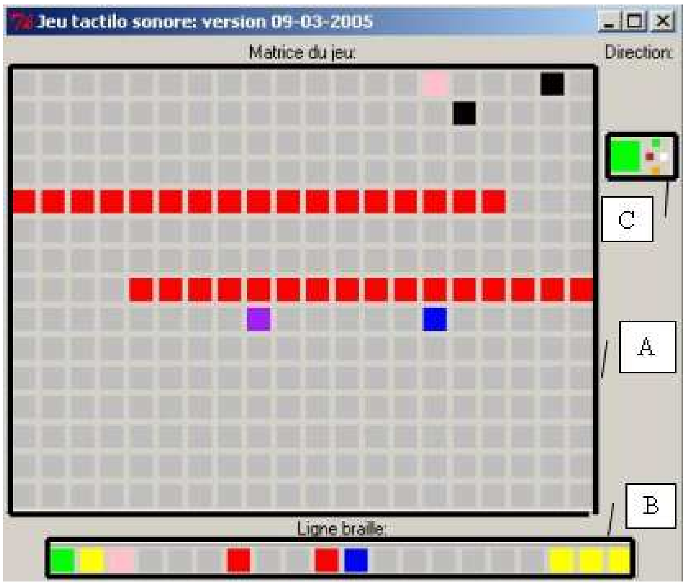


Figure 24: Maze Game interface

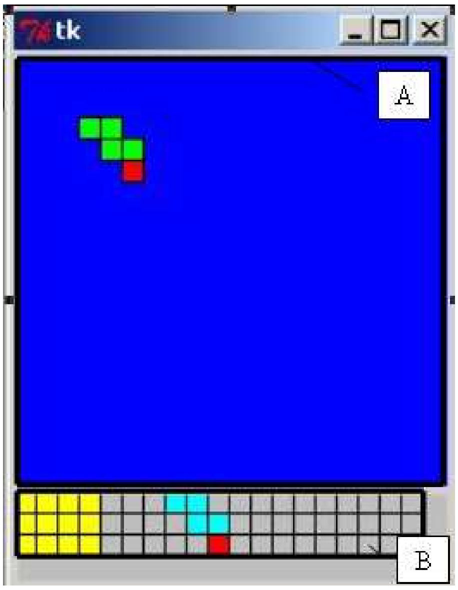


Figure 25: Snake Game interface

given by a colour. Currently, this game area is a no-toroïdal grid that is why a state "out of game" has to be used all around the game area to delimit it in the Braille representation.

B : The second area is the Braille line representation. It sums up the game area represented on the Braille terminal and the way they are represented.

C : The third one is the direction area. This information represents the direction of the character in the case of a forward, backward, pivot displacement. Nevertheless, Snake game displacement is a multidirectional one so this information is not useful and has disappeared.

Moreover, these interfaces are based on game and direction colour codes. The Maze one is the following: RED = Obstacle, YELLOW = Out of game, PINK = Exit, BLACK = Enemies, VIOLET = Object, BLUE = Character, GREY = Empty area. Whereas Snake one is: RED = Apple, YELLOW = Out of game, BLUE or GREY = Empty area, GREEN or CYAN = Character

Colour code are used to facilitate development but it can be easily replaced by images. It could be very interesting to use images to develop a prettier interface. Indeed, these games could be used as standard games and not only as accessible games. So children with or without visual impairment could share it and play together.

This notion of shared game is very important in accessible video games development. People with disabilities do not want to create a new video games community they want to share the same one than everybody.

3.5 Examples

In order to illustrate Braille representation, we will introduce an exemple of each game. From the two figures used to present applications interface (see figure 24 and 25), we can build the corresponding Braille representations.

In the case of the Maze game with a forward, backward and pivot displacement and a 1×20 or 20×1 window according to the character direction, we obtain the Braille representation of figure 26. In that case, the direction is north so the window used is a 20×1 i.e. the current column of the character. But the enrichment allows creating a 20×3 window. The first cell of the line gives this direction and the remainder of the line gives the neighbourhood. On the other hand, this line is centered on the character which explains that the second cell represents a sector in the state out of the game (in yellow on the interface).

And in the case of the Snake game with a mutlidirectionnal displacement and a 3×20, we obtain the Braille representation of figure 27. The window is a 3×20 one and the enrichment allows having a 5×20 one. In the same way, the window is centered on the head of the snake which explains that the first cell represents sectors in the state out of the game. Moreover, the displacement is a multidirectional one so direction does not appear.

3.6 Preliminary tests

3.6.1 Several kind of tests

Several kinds of tests have to be done in order to validate the games and associated Braille representations. These tests can be breaked down in three kinds:

- Tests to validate the functionning of the application and the building of the representation. This kind of tests is realised during the conception, the development of the application, and before to introduce it.
- Tests to validate the Braille representation. This kind of tests can be realised by two kind of people:

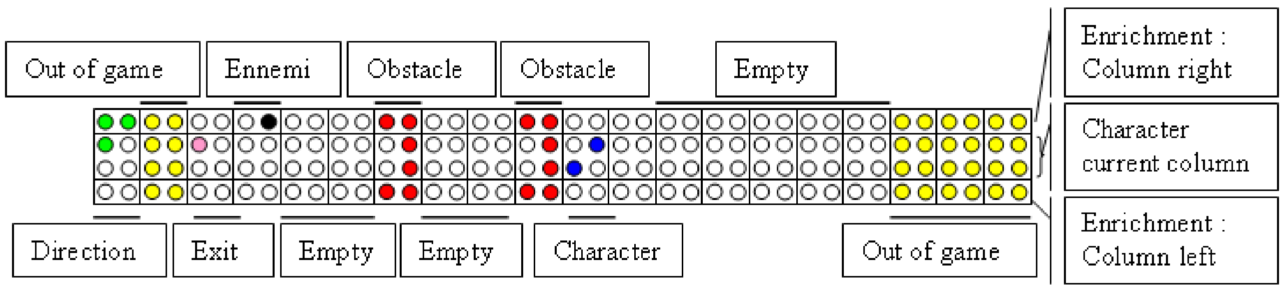


Figure 26: Maze game - Exemple of Braille representation

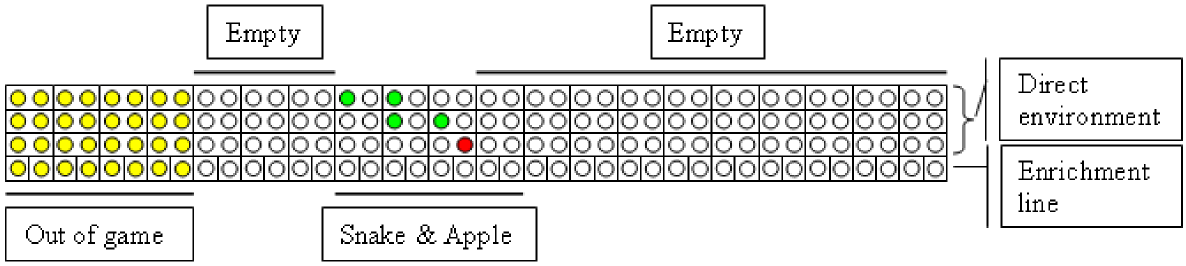


Figure 27: Snake game - Exemple of Braille representation

  - people without visual impairment but having a knowledge about it i.e. people working with visual impaired persons, medical personal, ...
  - advanced blind computer science users. These persons are adults and have a large knowledge of computer science so they understand the representation faster and they know computer science limits or abilities.

  Finally, the knowledge or/and the experience of visual impairment allows assessing the Braille representation. Then, the best points will be kept to build better representations.

- Tests to validate the playability of such games. This last kind of tests have to be done with young blind persons because these games are intended for them.

3.6.2 Tests to validate the functionning

This kind of tests has been made all along the development of the aplications and before to introduce them. Developers were not visually impaired people and did not have a large knowledge about this impairment. To realised these tests we have used several methods:

- We have played only with the Braille line representation on the interface (see figure 24 and 25 - area B)
- We have visually played from the Braille terminal. Indeed, the different configurations of splinters can be seen and not only felt with the fingers.

3.6.3 Tests to validate the Braille representation

This kind of tests has been made thanks to an irish advanced Braille reader.

The first conclusion drawn from these tests is the craze for this kind of games. Indeed, he was very happy to take part in these tests and play from the Braille terminal.

The second one, and the most important one, focuses on the playability of the games. After a few minutes to explain the functioning of each game, he played without any problem. Moreover his experience of Braille-reader allows assessing the different representations. According to him, a good representation has to rally the following points:

- The character has to appear directly or not in the Braille line but his position has to be always the same.
- The reading mechanism of a Braille-reader is the following: the reader put his two index fingers on the Braille cell associated with the character. Then, he slides his two index fingers simultaneously from the character to exterior of the Braille line (see figure 28).

These two remarks permit to select several representations as line or column ones and suppress others as cyclic route because character does not appear. A new representation has been proposed. It is based on a 3×3 window (see figure 29) with the following route (see figure 30):

In this representation, without enrichment, character position is given by the two cells which translate the elements behind and front of him.

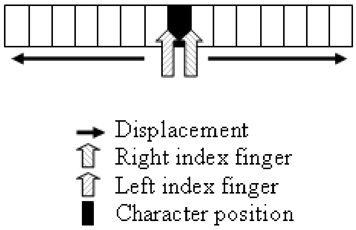


Figure 28: Braille reading mechanism

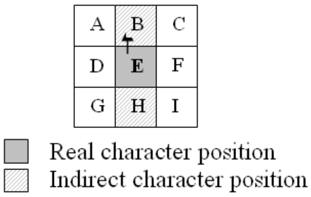


Figure 29: New Braille representation - 3x3 window

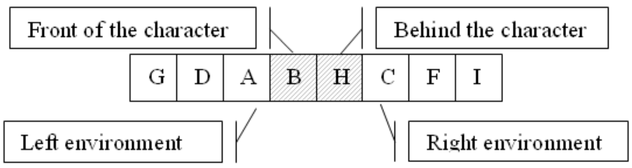


Figure 30: New Braille representation - route

Then, the third conclusion drawn concerns the enrichment. Indeed, it can be useful for an experienced player but not for a beginner. For example, he did not use it immediately and it can be a problem since these games are designed for children who are not as good Braille reader. So, enrichment presence as to be validated with tests with children.

Nevertheless, these conclusions can not be considered as universal. Indeed, only one visually impaired player has tested our games so new tests have to be done to confirm our results.

3.6.4 *Tests to validate the playability*

Finally, the last kind of tests, which has to be done, are tests with children. Today, we are looking for young Braille reader to realize such tests and validate the games.

4. BLINDGAMES PLATFORM AND SEMI AUTOMATIC GENERATOR

4.1 BlindGames platform

The preliminary tests permit to draw another conclusion : a platform to start the different games is necessary. Indeed, during these tests with a visually impaired person, we have to help him to start the games. But, to be accessible, an application has to be accessible from its launch to its end.

So we have decided to develop a platform called BlindGames. This platform offers a simple way to access the games but it also allows accessing to a semi-automatic generator of such games.

4.2 Semi automatic generator

4.2.1 *Games internal structure*

The idea of a semi automatic generator comes from the structure of the two games developed.

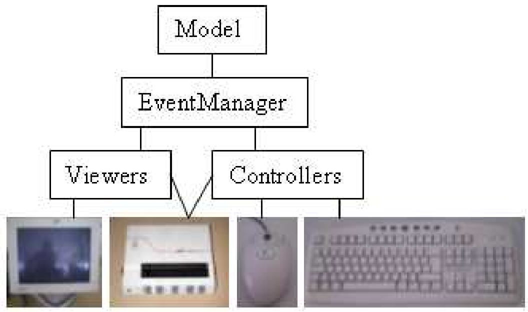


Figure 31: SJBROWN model - general structure

Indeed, each game presents the same structure. This structure is based on ”sjbrown” representation [10] which general structure is the following (see figure 31)

The different elements of this structure are:

- ”Viewer”: They permit to show the state of the model and its different elements on the output devices (screen and Braille terminal here but we can think about other kinds of equipment). These viewers can be used similtaneously or not according the necessity to have a sharable or a dedicated game. In our works we focus on sharable games.
- ”Controller”: They permit to acquire the different events from the input devices (as keyboard, Braille terminal, mouse, ...) or the internal events (as timer, ...) and to convey them to the EventManager. In our works, the displacement of the character can be executed from the Braille terminal or the keyboard. Indeed, the Braille terminal gets, in addition to the Braille cells, several control keys. For exemple, with the ECO ONCE 20 Braille terminal (see figure 1), there are five keys on the front of the equipment.
- ”Model”: It contains all the information linked to the structure of the game as : the game area, the character and his position, the ennemis, ...
- ”Event Manager”: This structure is the heart of the application. Its role is to take into consideration the different events from the ”Controllers” and to send them to the ”Model” (to save the modifications: character position, ...) and to the ”Viewer” (to update the interface)

This scheme has different interests. First, it takes several controllers and several viewers into consideration. This particularity allows using keyboard, mouse and Braille terminal (as input devices) or screen and Braille terminal (as output devices) simultaneously.

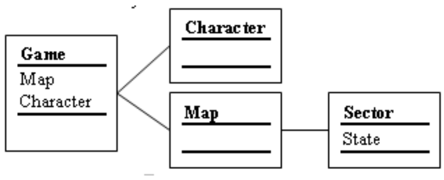


Figure 32: SJBROWN model - general scheme

Moreover, the model scheme (after some simplifications) represents game area as a map made up of sectors and where each sector has a state. This representation is good way to develop games which are based on grid area (see figure 32).

Finally, the different applications have the same internal structure but also very similar interfaces. From this remark, we have decided to create a semi automatic generator.

This semi automatic generator permits to create a skeleton of Python game sources from several forms. It is called semi automatic because programmer keeps having to finalize the application but his work is clearly facilitated.

4.2.2 *Semi automatic generator forms*

Semi automatic generator is based on the use of simple forms. Each one permits to get information on the game to build. They are based on four steps.

1. The first step permits to give general information about the model like his name, the map size and the use of enemies or obstacles, their position (see figure 33).

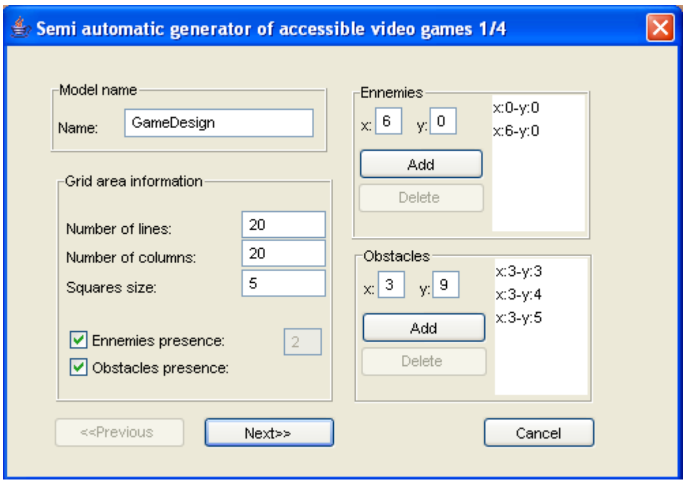


Figure 33: Semi automatic generator form - step 1

2. The second one is associated with the displacement asking the kind of displacement, the use of a timer to move and the direction display (see figure 34).

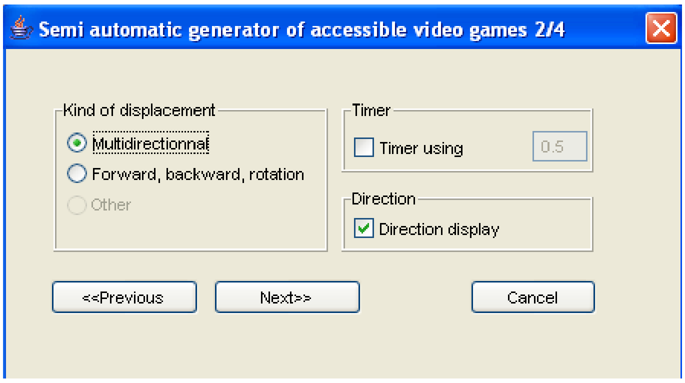


Figure 34: Semi automatic generator form - step 2

3. The third one is linked with the Braille representation information as the shape of the window (number of lines and columns), the number of splinters associated to direct and indirect neighbourhood representation (see figure 35).

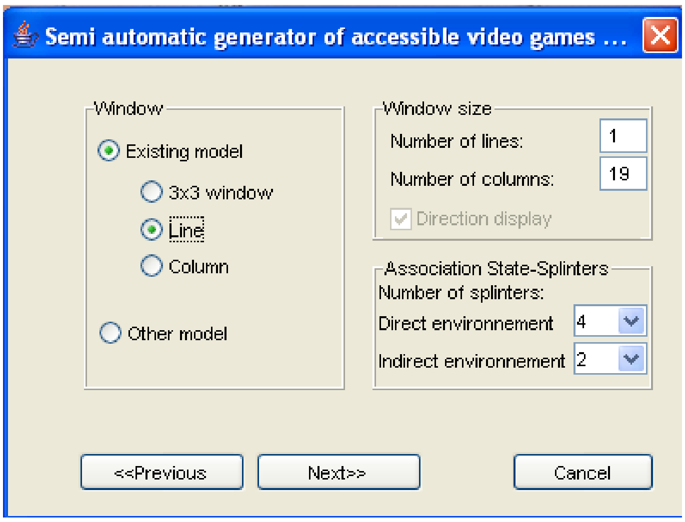


Figure 35: Semi automatic generator form - step 3

4. Last, the fourth form permits the management of the different states possible with the association of the configuration of splinters and the behavior when the character meets this state during a displacement (see figure 36).

Finally, from few information given about the game wished, the semi automatic generator build the Python skeleton of the game.

So, this semi automatic generator permits to facilitate the creation of new games but also new representations. There is just one constraint about games created: they have to be based on the displacement of a character through a grid.

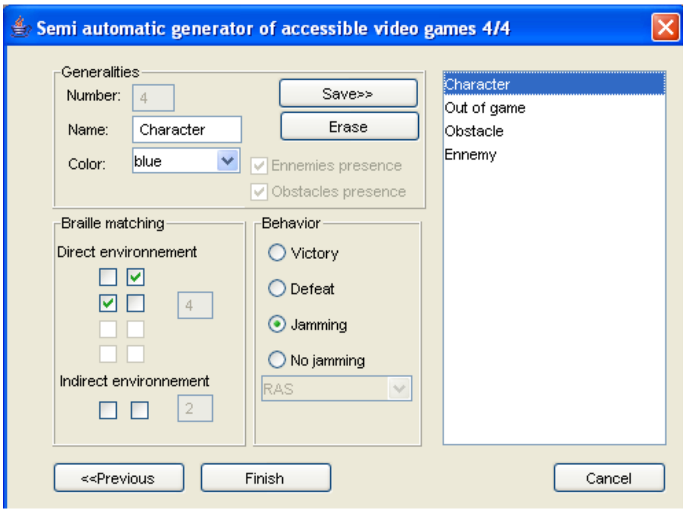


Figure 36: Semi automatic generator form - step 4

Moreover, the platform permits to manage the game all along its creation from the launching of the forms to its validation. Indeed, the skeleton generated by the semi automatic generator has to be finalized to obtain a playable game. The platform permits to facilitate this finalisation proposing tools to edit the skeleton or to test the game.

5. IMPROVEMENTS

Many improvements, more or less important, can be added to this project.

The first one, and the most important, is the use of sounds. Sounds will add a new dimension to these games. They will permit to translate more information and facilitate spatial representation.

Next, the use of an unlimited map can be interesting to develop other kind of games as car races, ...

Then, the generator improvement to facilitate more and more accessible games creation represents also a major enrichment.

On the other hand, downloaded accessible video games are often in one language or in English. Nevertheless, if many adults speak English, children do not, so a multi-languages platform has to be developed to allow all children to play these games. So our platform has to be improved to take others languages into consideration.

Finally, developed games are only single player games. It could be interesting to experiment multi players games to increase the share idea between children.

6. CONCLUSION

Today, there are just few accessible video games for visually impaired persons and especially for children.

Moreover, existing games are often based on sounds but rarely on Braille terminal. Nevertheless, the learning of Braille begins with a familiarisation with the Braille terminal to increase touch sensibility.

In this work we have described several accessible games and also a tool to easily create new ones. Moreover, the aim of this tool is to reduce programming work and, why not, to suppress it. So, people, who just have some notion in computer science, as medical personal or educator will be able to develop their own games and especially their own representations.

Then, many improvements could be made as the enrolment of audio information, the evolution of the semi automatic generator to limit user intervention, the development of other kind of games, etc.

7. ADDITIONAL AUTHORS

M. Slimane mohamed
Laboratoire d'Informatique
Université François Rabelais, Tours, France
email: `mohamed.slimane@univ-tours.fr`

8. REFERENCES

[1] BrailleNet Web site
http://www.braillenet.org

[2] TiM project documentation *Adaptations multimodales de jeux multimédia*
Dominique Archambault, Aurélie Buaud and Sébastien Sablé
http://www.snv.jussieu.fr/inova/tim

[3] Arnaud Puret, Dominique Archambault, Nicolas Monmarché, and Mohamed Slimane. A simple game generator for creating audio/tactile games. In *Assistive Technology: From Virtuality to Reality (AAATE'2005)*, Lille, France, September 6-9 2005.

[4] Libbraille documentation *Libbraille: a Portable Library to Easily Access Braille Displays*
Sébastien Sablé and Dominique Archambault
http://libbraille.org/

[5] Web site specialized in Handicap
http://www.yanous.com
and several articles about accessible video games:
http://www.yanous.com/tribus/aveugles/aveugles040319.html
http://yanous.com/logiciels/essais/essais030321.html

[6] Level Games Web site
http://www.levelgames.net
http://www.arcess.com

[7] Wheels Web site
http://www.rjcooper.com/wheels/

[8] Tachido Web site
http://www.novascoop.com/article.php3?id_article=4192

[9] JeuAccess Web site
http://www.jeuaccess.com

[10] SJBrown representation Web site
http://sjbrown.ezide.com/games

4.2 [CGames'2007]

Titre	**Tampokme : a Multi-Users Audio Game Accessible to Visually and Motor Impaired People**
Auteur(s)	Thomas Gaudy[(a)], Stéphane Natkin[(a)], Cécile Le Prado[(a)], Thierry Dilger[(a)] et Dominique Archambault
Affiliations	[(a)]CNAM, Paris
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of CGames'07 (11^{th} International Conference on Computer Games : AI, Animation, Mobile, Educational & Serious Games)
Editeur(s)	Qasim Mehdi, Pascal Estraillier et Michel Eboueya
Lieu/Date	La Rochelle, Novembre 2007
Pages	73–78

Tampokme – The Audio Multi Players One Key Mosquito Eater – a été développé à l'occasion du concours *"The 2006 DonationCoder.com Accessibilty Game Programming Contest"*[8]. Les jeux participants à ce concours devaient être accessibles ou bien par des personnes handicapées visuelles en utilisant seulement le son, ou bien par des personnes handicapées motrices en n'utilisant qu'un seul bouton. *Tampokme* répondait à la fois aux 2 catégories. C'est un jeu multi-joueur permettant à 4 joueurs handicapés visuels, handicapés moteurs ou valides de jouer ensemble selon des principes à la fois compétitifs et collaboratifs.

Tampokme est aujourd'hui distribué par *Ceciaa*. Il est disponible à l'adresse suivante :
`http://braille-vocal.ceciaa.com/jeux-sonores-c319-155.php`

Voir mémoire section **3.2.2**, page 35

[8] `http://www.donationcoder.com/Contests/agame`

TAMPOKME : A MULTI-USERS AUDIO GAME ACCESSIBLE TO VISUALLY AND MOTOR IMPAIRED PEOPLE

Thomas Gaudy(1) Stéphane Natkin(1&2) Cécile Le Prado(1&2) Thierry Dilger(2) Dominique Archambault(3)

(1) Centre De Recherche
en Informatique du CNAM
292, rue St Martin,
75003, Paris, France
tomgody@yahoo.fr
natkin@cnam.fr
cecile.leprado@free.fr

(2) ENJMIN/CNAM
121 rue de Bordeaux, Angoulême,
France
dilger@studiowan.com

(3)Université Pierre et Marie Curie,
Inserm 483/Inova
9 quai Saint Bernard,
75 252 Paris cedex 5, France
00 331 44 27 26 10
dominique.archambault@upmc.fr

KEYWORDS

Game Design, Accessibility, Audio game, Visual impairment, Mobility disability.

ABSTRACT

We present in this paper a quick overview of the games accessible to the people who are visually impaired. First, these audio games require often a too great number of commands: for this reason, they are hard to learn and many people do not enjoy playing them. Second, the users want more audio games with multiplayer feature. This paper presents the audio game TAMPOKME, which is accessible to people who are visually and motor impaired. It can be played from one to four players and combines principles of competition and cooperation. We describe in this paper the methodology that enables us to produce such a game. After presenting the aims of TAMPOKME, we detail its interface and gameplay. We then present how this audio game has been improved, in a sound design perspective, by the students of the Graduate School of Games and Interactive Medias, France, and the interface qualities that have resulted from it. It is possible to make an audio game with very simple control. In that way, such a game can be easier to learn and it can be enjoyed by another community than the visually impaired one.

STATE OF THE ART

Research carried out on video audio games, accessible to people who are visually impaired, focuses both on the accessibility of the man/machine interfaces and the question of an adapted gameplay. In this paper, when we mention video games, we refer to « classical » commercial games and with the expression audio games, we refer to video games specifically designed for people who are blind. This paper results from a state of the art concerning this type of media (Gaudy et al.2006a), the main results of which are the following:

There are over 400 video audio games accessible to blind people and there are more and more developments. These games follow the principles of « classical » video games, but with a limited cost, and, as a result, a limited content and complexity. However, the sensations given by these games are quite different from those given by video games. They are, at first, quite disconcerting for sighted players because of the absence of a visual interface and of an unusual appeal to hearing. The aims of these games often seem confusing to them, and the interaction, awkward.

We consider that this difficulty to enjoy audio games is related to the excessively great number of commands the players have to use. A great number of audio games requires more than ten keys on the keyboard. In consequence, the player has to pass through a learning process, first by reading the leaflet and then by playing the game. Most of the time, players have to read the leaflet again and try the game many times before they can begin to enjoy it. Another consequence of this long and unavoidable learning phase is that people who can discover other interactive games in a more easy way will not try these audio games. We think that it is for this reason that sighted players do not want to play audio games.

We experienced several ways of improving them, while developing a maze audio game. In particular, the learning of the perceptive interface has to be progressive with a very pleasant and easy starting situation. In that way, people can play our audio game without having to read the leaflet. We think that this approach can be implemented in a more efficient way: as videogamers, audio gamers should play simple games without the help of any kind of verbal instruction. We have obtained some encouraging result on that purpose (Gaudy et al.2006b).

USER'S NEEDS

In parallel to the state of the art, we inquired on the needs of young visually impaired people concerning digital leisure thanks to a questionnaire and interviews. The main results of this survey seem to show that the games that have been specifically designed for them are not necessarily their favourite ones. The complexity of the instructions given in a foreign language constitutes a significant factor of rejection. As most of the audio games are in English, they are little played by non English-speaking visually impaired people. Furthermore, numerous video games, supposedly inaccessible to disabled people, are, in fact, much enjoyed by many players who are visually impaired or partially sighted. According to another survey carried out in 2007, 64% of the interviewed visually impaired people said that they liked playing « ordinary » video games (France 2007). Fighting games are much enjoyed, as well as some race games. We met visually impaired players who finished very complex video games such as « Zelda, a Link to the past » or « Donkey Kong Country ». The strategies used to play these games are amazing and require ingenuity, perseverance, and a good ability to memorise. It seems that the pleasure they have to play these video games is not the unique motivation of the people who are visually impaired. The fact that they practice a leisure activity intended for the general public also plays a significant role. As a matter of fact, these games can be considered as vectors of communication between sighted and visually impaired or partially sighted people. They therefore have a deep social importance. This social feature has been noticed as being particularly important for multiplayer video games (Manninen, 2003)

This particular feature is particularly required by the editor of "Audyssey", a specialised magazine about audio gaming, who has followed the evolution of audio games since ten years (Feir, 2006). The future audio games will need to include more social features or multiplayer modes. At this point, we wanted to work on this original contribution: a multiplayer feature on a new original audio game. We work on this aspect of audio games by considering also the problems linked to controls that are too complex.

QUICK OVERVIEW OF OUR GAMES

In the continuation of those observations, we developed three audio games. The first one is a musical maze game and the second one is an exploration audio game which requires the mouse to provide different feelings to visually impaired players. In audio games, the mouse can no longer be considered as an inaccessible device for visually impaired people (Bors, 2007). In those two games, we want to limit the number of required keys. In that way, we hope they are easier to handle. We had the opportunity to test them on a man with mobility disability at an exhibition. The maze game in particular was much enjoyed. The other one is not accessible to motor impaired people due to the mouse. The maze game only requires the four arrow keys. If one more key was required, this game could not be accessible to that player. Since then, we have considered the possibility to make audio games which can be played by people with motor impairment.

Other studies on audio games pay attention to multi handicap problems and there are games that can be played by visually impaired people or by motor impaired people. Those produced by UA-Games of the ICS HCI Lab are impressive and give encouraging examples of strong accessibility applied to action or puzzle games (Grammenos et al. 2005 ; Grammenos et al. 2006 ; Grammenos et al. 2007).

Our third game, which this paper deals with, is called. TAMPOKME : « The Audio Multi Players One Key Mosquito Eater ». TAMPOKME was developed for a competition called « DonationCoder Game Accessibility Contest 2006 » organized by DonationCoder and by the International Game Developer Association (IGDA). This competition offered to develop a game in one of the two following categories: either an audio game accessible to visually impaired people, or a game with a single switch accessible to people with motor disability. TAMPOKME tries to meet the two constraints. It is a game that can be played by one to four players and that combines principles of competition and cooperation. It is the first game to combine accessibility for visually impaired people and physically impaired people and with a multiplayer mode which allows up to four players to play together at the same time.

In the following paragraphs, we detail the aim of this study, the methodology and the evolution of TAMPOKME. This game was then improved by students of the Graduate School of Games and Interactive Medias, Angoulême, France (ENJMIN). At last, we will describe the remaining problems of the game.

AIMS OF THE STUDY

The development of the game TAMPOKME is part of a research on the understanding and the development of the interaction principles specific to visually impaired people but also, with this game, to people with mobility disability. This development is part of a current of research carried out by « Game Accessibility Special Interest Group » of IGDA (IGDA 2004), the Game Accessibility Project, the Audio games.net community and the French project TIM (Archambault 2004).

TAMPOKME is an experiment on several aspects of the conception of an interface and a gameplay based on sounds, usable with a single switch, and intended for users with different modalities of perception: the game must be playable for visually impaired players, physically impaired people and people without these kinds of disabilities.

The aims of the study are simple. First, we want to develop such a game. Second, we want players to find this kind of game enjoyable. We will explain in this study that the first goal was very simple to reach, but the second one was much harder. Many tests with blind and sighted players show a lot of problems. The experiment carried out with the students of ENJMIN enabled us to note that it was possible to improve the fun mechanisms of the interaction with the improvement of the sound design. This paper will describe the development process and in particular the significant help provided by the ENJMIN which has allowed us to bring a new and original pleasant free audio game.

METHODOLOGY

We wanted to develop a very simple game. TAMPOKME is a reflex game. The players have to identify a sound signal and properly react by pressing a key the appropriate number of times.

From this point of view, the rules are not complex at all. We wanted to make a simple game because we wanted to have the maximum time for the consultation and evaluation process by the target user group. This particular aspect of development takes more than three months with more than twenty sighted players and more than ten visually impaired players. We have no performed test with motor impaired people and we assume that this was a mistake.

The development method of this game is quite classical: we have followed a spiral cycle through successive prototyping, having the functionalities evaluated (Natkin 2006).

Each test requires only one new tester, and eventually other players which are already familiarised with the game, on a new version of the game. We consider each time the whole behaviour of the tester. With this approach, we could not focus on specific results as the time or the progression in the game. We wanted to identify when the tester feels difficulties, what was the nature of these difficulties and what in the game makes them occur. In that way, each test reveals a lot of problems and was followed by improvement developments.

This kind of development process presents a great advantage: one tester is sufficient to give many hints on the needs for improvement. But it presents also a great disadvantage: the results are just hints, based on the interpretation of the tester's behaviour. It is not a statistical and objective type of result. In consequence, at this point of the development, we can't prove the validity of the game. We assume this is a major weak point of this study but we will not only describe the qualities of TAMPOKME. We will also insist on the negative aspects. In that way, we hope this paper will contribute in an easier process to future works and research in these fields of studies.

EVOLUTION OF TAMPOKME BEFORE THE ENJMIN'S HELP

The first test of the first prototype revealed that sighted people did not understand the rules of the game because they are not familiarised with games without screen. The solution for this problem was to implement a « scenario ». The « scenario » of the game is based on a story of a carnivorous plant, controlled by the player, and different kinds of mosquitoes which come in turn to attack the plant. "Regular mosquitoes" require the key to be pressed once whereas "super mosquitoes" require the key to be pressed twice and "toxic mosquitoes" require no pressing of the key.

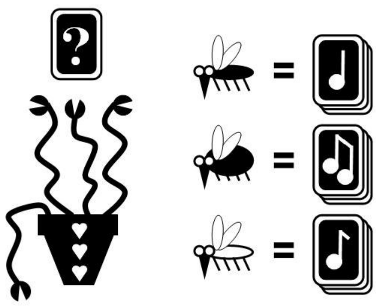


Figure 1. Representation of the principle of the game following the analogy with the game card. Each player controls a stem. Each kind of mosquito, "normal", "super", "toxic", has its own type of sound, produced in various versions.

Each kind of mosquito is associated to a type of sound signal and each sound signal can be produced in a variety of forms, in order to give the game variety and also more difficulties, as the number of sounds to identify increases. In fact, to make an analogy with a card game, it is as if a card was picked up at random and that the players had to determine what kind of card it is as quick as possible (see Figure 1).

We had to work on the difficulty of the game. The testers familiarised with other audio games found this one too easy. We realised that the different sound signals have to sound quite alike. The process of recognition is willingly made ambiguous and difficult.

As the game requires only one key, it was possible to include a multiplayer dimension more easily. Each player uses his/her own key on the same keypad. The sound signal calls for all the players to react together. Then, we had to improve this multi-player dimension. The test reveals that the social aspect of the game is more pleasant if the multiplayer mode is both cooperative, as the game is over only when all the players lose, and competitive, as the players can compare their performances through their score. These competitive and cooperative dimensions are also found through specific "life point" systems, a principle which is found in most video games. We make a distinction between individual life points, for the competitive dimension, and community life points, for the cooperative dimension. Concerning the competitive life point system, each player has a life point that he/she loses if he/she makes a mistake, and is thus temporarily withdrawn from the game. If all the players lose their life point, it is a community life point that is lost and all the players take back their individual life point. If all the community life points are lost, the game is over. The cooperative dimension is the most important one as the best player does not play against the worst one, who, thus, can play longer.

In that way, the game had a better social feature, but the rules of the game became more complex, and we had to improve again the scenario. As the information of the game is only based on sound, we had to manage it temporally and not spatially, as it is the case with « classical » video games. All the information cannot be given simultaneously. We made a scheme of the different steps that the game presents in turn and that a beginning user has to go through to know how to master the game (see Figure 2). All these steps are explained to the players with recorded verbal instructions included in the game. The time needed to explain the rules through the scenario became longer and we had to adjust again the difficulty. We decided also to implement a shortcut in the game for players who are familiarised with TAMPOKME. At the end of the game, a last manipulation is explained, which has to be done at the beginning of the next game: in that way, the player can go to the last step directly without going through the intermediary steps.

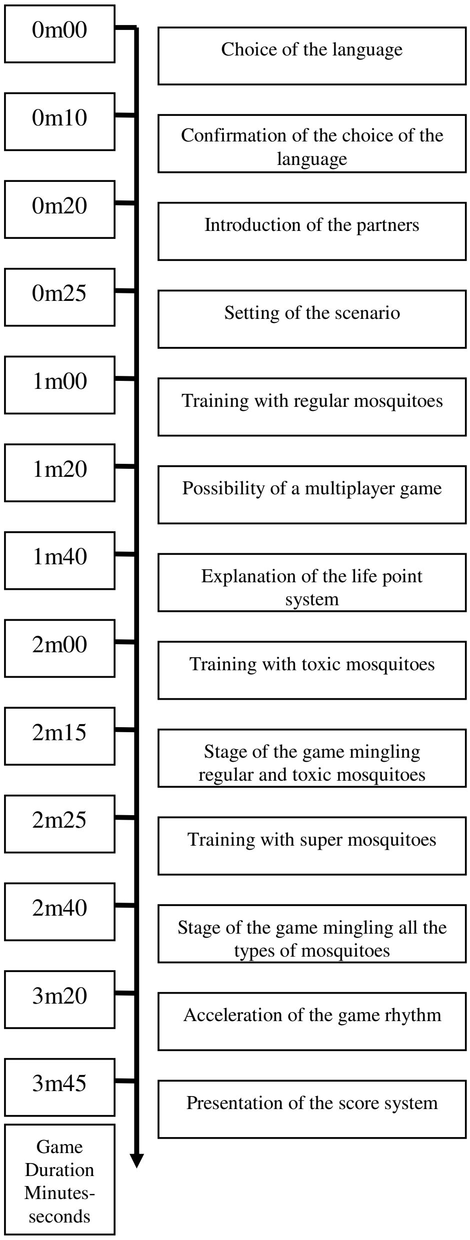


Figure 2. Order of presentation of information.

Then we had to consider two possible kinds of final users: the French speakers and the English speakers. We developed a one-key audio system permitting to have a choice between two languages: English and French. The game alternatively offers to play with either French or English instructions. The player has to press the key at the right time to choose the language. The game then asks the player, in the corresponding language, to press the key three times to confirm his/her choice and to continue the game. In an incomprehensible language, the player would have little chance to do the right number of interactions, and wrong manipulations are punished with a new change of language. Specific sounds punctuate the number of interactions: neutral sounds when the number is not sufficient, a more melodic sound when the required number is reached and a more aggressive sound when the number of interactions is too high. This type of sound system reinforces the comprehension of the interactivity of this audio menu.

At this point of development, the game still had some perceptive defects and defects of interface. Three categories of sounds can be essentially distinguished: the sounds emitted by the plant - swallowing, digestion noises, super swallowing, produced by a double press, the sounds of mosquitoes and the verbal instructions. Each category had specific problems in that version. The verbal instructions that were meant to give a context to the action were too long. The flow of instructions was slightly accelerated so that the beginning of the game is not too long, even though the instructions are consequently less intelligible. If the tests that were done with visually impaired people did not reveal any problems, on the other hand, the comprehension of the game instructions was often confusing for most of the sighted players. The problem of the comprehension of included in-game verbal instructions by the sighted people seems to be recurrent with most of audio games. Finally, the verbal instructions were given by three characters: a mad scientist, the carnivorous plant and a voice off. The quality of the voices made it difficult to distinguish the different characters. As for the sound of the mosquitoes, they were too aggressive. We used a synthesiser of which the sounds were a little too stressing to make the game pleasant. At last, the noises of the plant were totally abstract and did not help the player to have an idea of the situation of the game. Generally speaking, the game triggered a great hearing tiredness. The students of the ENJMIN, whose major is sound conception, helped to improve the sound design.

SECOND EVOLUTION PRODUCED WITH THE STUDENTS OF ENJMIN

In the ENJIM school, six majors are offered to the students: game design, sound conception, image conception, project management and ergonomics. The four first year students, whose major is sound conception and who studied either music or sound engineering for their graduation, have, as a practical exercise for the sound design course, to improve the game TAMPOKME. They were helped by two students whose major is Game Design.
This period of development combined two phases. In the first one, the students, working in pairs, suggested three very different versions of TAMPOKME.
The specified aim was clearly to solve the identified problems:
a) The sounds of mosquitoes that created an atmosphere too irritating to be pleasant
b) The sounds of the different players which had so different characteristics that it was difficult to picture that they all referred to the same musical entity.
c) The verbal instructions are quite rough and probably lack a musical design.
The overall aim was to create a coherent, pleasant and efficient world through the sound design.
The students started by testing the game, analysing it and suggesting, from what has been noted above, what could be improved from the player's viewpoint. They then thought

about the possible transformations of the sounds of the plant and the mosquitoes and proposed what they wanted to do with the sounds before incorporating them into the games.
Then, in a second phase, we consider the works of the students in a new iterative process with new testers. The kept modifications exclusively concern the change of the sounds, of the atmospheres and the music, as well as the substitution of some locutions with evocative sounds. They then suggested some ideas to improve the interface and the gameplay: the use of sound spatialisation (Gonot et al.2006), a multiplayer mode on several machines, a system of gradual difficulty by changing the sound aesthetics of the game for each level.
All the suggestions were very interesting, but all of them could not be materially applied. We focused on the three sound versions. The game, as it was designed, could not include all the sounds: the limit was three kinds of mosquito sounds, which remained unchanged from one game turn to the other. After some thinking, it appeared that the life duration of the game could be expanded if the game randomly chose three categories of mosquito sounds from a wider range of mosquito sounds for each new game turn. In that way, we could better use all the students' works and from one game turn to the other, the sounds changed, which means that the challenge is renewed every time. Unlike the first version, the players can no longer rely on their learning and recognition of the sounds of the previous game turn (see Figure 3). The game becomes more difficult, which is not bad, given that, after a few game turns, visually impaired players could very quickly manage not to make any error in the first version.

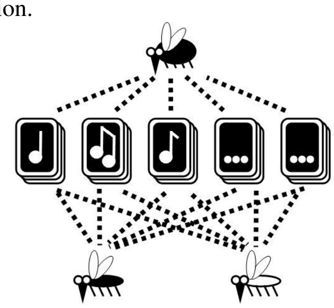


Figure 3. Visual representation of the choice at random of the categories of sounds attributed to the different kinds of mosquitoes at the beginning of the game, inspired by the works of the students of ENJMIN.

GOOD AND BAD ASPECTS OF TAMPOKME

Since the distribution of the game TAMPOKME, we have been receiving numerous comments. They are generally very favourable. Sighted people, visually impaired people and physically impaired people managed to play together on the same game and a lot of them enjoy it. Comments about the game can be found on the Internet.
We will now consider three actual limitations of the game.
First, the difficulty of the game is still not good enough. From a subjective point of view, the difficulty of the game is not the same for the different populations of gamers. Sighted players find Tampokme quite hard to master and to understand whereas visually impaired players find it too easy at the beginning. For motor impaired players, the difficulty has to be adjustable because some people have a very huge handicap and it is harder for them to react within the good timing.
Second, the multiplayer mode and the solo mode both have to be improved. The game is pleasant when it is played by two or three players, but it presents less interest when only one player tries it. At last, when the game is played by four players at the same time, the situation can be confused and sometimes, players did not manage to recognize the sounds produced by their own interaction.
Third, the accessibility of the game for the multiplayer mode presents a problem for the motor impaired people. The keyboard can hardly be used by four people with this type of handicap. The mouse should be used. If physically impaired players had been included in the development process, this problem would not occur in the final version.

CONCLUSION

Many projects start with complex features on the paper and the iterative process which requires to simplify them. We have presented the development of an audio game, which was very simple at the beginning. This point allowed us to modify the game in a more complex way in order to take into consideration the needs of the testers. Finally, we succeeded in developing a new audio game which provides a strong social feature by gathering people with varying disabilities in the same game environment. The controls required by the game are simple and the players do not have to read the leaflet, in spite of a relative complexity of the actual rules. The experience of students in a video game school was particularly useful to obtain a more pleasant game. However, the game is not perfect. Our next aim is to solve the three main actual problems.

ACKNOWLEDGMENTS

We thank Nicole Chun Lu, Fabien Deleurme, Samuel Lachaud, Jean Leman, Xavier Montels and Bertrand Poulain, students of the 2006-2007 promotion of ENJMIN for their help.
Thanks to all the testers and the players.
This work has partially been realised with the support of CECIAA, in the framework of a CIFRE contract.

REFERENCES

Archambault, D. 2004. "The TiM Project: Overview of Results". In *Procedings of the ICCHP 9th International Conference on Computers Helping People with Special Needs (Miesenberger, K., Klaus, J., Zagler, W., and Burger, D. editors), Lectures Notes in Computer Science* (Berlin, Germany). Springer. 248-256.

Bors, B. 2007. "The Role Of The Mouse In Audio Games". *Resources of Game-Accessibility.com from the Bartiméus Accessibility Foundation.*

Feir, M. 2006. “A Decade’s Accessible Gaming: An Old Editor’s Anniversary Observations” *Audyssey magazine* 49.

France, M. 2007. “Audio Games Survey Results”. *Audyssey magazine* 50.

Gaudy, T., Natkin, S., and Archambault, D. 2006. “Classification des jeux sonores selon leur type de jouabilité ». In *Proceedings of Handicap 2006 Conference* (Paris, Fr, Jun. 7-9). IEEE, 221-226.

Gaudy, T., Natkin, S., and Archambault, D. 2006. “Playing audio games without instructions for uses: To do without instruction leaflet or without language itself ?”. In *Proceedings of CGAMES 2006* (Dublin, Ir, Nov 21-24). IEEE, 263-268.

Gonot, A., Natkin, S., Emerit, M., and Château, N. 2006. “An Experiment in the Perception of Space through Sound in Virtual World and Games”, In *Proceedings of CGAMES 2006* (Dublin, Ir, Nov 21-24). IEEE. 274-278.

Grammenos, D., Savidis, A., and Stephanidis C. 2005. “UA-Chess: A Universally Accessible Board Game”. In *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction* (Las Vegas, Nevada, July).

Grammenos, D., Savidis, A., Georgalis Y., and Stephanidis C. 2006. “Access Invaders: Developing a Universally Accessible Action Game”. In *Proc. ICCHP 2006 10th International Conference on Computers Helping People with Special Needs* (Linz, Austria, July). 388-395.

Grammenos, D., Savidis, A., Stephanidis, C. 2007. “Unified Design of Universally Accessible Games: Universal Access in Human-Computer Interaction, Applications and Services.” In *Proceedings (Part III) of the 4th International Conference on Universal Access in Human-Computer Interaction* (Beijing, China, August). 607-616.

IGDA. 2004. "Accessibility in Games: Motivations and Approaches". White_Paper.

Manninen, T. 2003. “Interaction Forms and Communicative Actions in Multiplayer Games”. *The international journal of computer game research* 3, no.1.

Natkin, S. 2006. “Video games and Interactive Media: A Glimpse at New Digital Entertainment”. AK Peters, AK Peters.

BIOGRAPHY

Thomas Gaudy studied at the National School of Video Game and Interactive Media (ENJMIN – France) and is now preparing a PhD entitled “Video Games for visually impaired people”. He has been developing four audio games which he is testing with users: two musical maze games, another one which is playable with a mouse, while the fourth one combines accessibility for visually impaired people and for people with motor impairment.

4.3 [TOE'2009]

Titre	**Pyvox 2 : an audio game accessible to visually impaired people playable without visual nor verbal instructions**
Auteur(s)	Thomas Gaudy[(a)], Stéphane Natkin[(a)] et Dominique Archambault
Affiliations	[(a)]CNAM, Paris
Type	Revue Internationale, avec Comité de lecture
Publication	Transactions on Edutainment II, LNCS 5660, Springer, Berlin
Editeur(s)	Zhigeng Pan, Adrian Cheok, Wolfgang Müller et Abdennour El Rhalibi
Volume/Date	Vol. II, 2009
Pages	176–186

Voir mémoire section **3.1.3.6**, page 33

Pyvox 2: An Audio Game Accessible to Visually Impaired People Playable without Visual Nor Verbal Instructions

Thomas Gaudy[1], Stéphane Natkin[1], and Dominique Archambault[2]

[1] Centre de Recherche en Informatique du Cnam
292, rue St Martin, 75003 Paris, France
tomgody@yahoo.fr, natkin@cnam.fr
[2] Inova/UFR919, University Pierre et Marie Curie-Paris 6
4, place Jussieu, 75252 Paris Cedex 5, France
dominique.archambault@upmc.fr

Abstract. In games, we can discern two approaches to learn how interactivity works: the instructions for use and the interactivity itself. The number of spoken languages is evaluated at more than six thousand eight hundred: for this reason, instructions for use can't make games understandable for all potential users, which is especially true for audio games accessible to visually impaired players, since those games can not count on visual support and have small budgets. Such games don't provide translation, perhaps because of a lack of cost effectiveness. So, if the purpose of a game is to learn in a friendly but challenging way how interactivity can become complex, why not start this process from the very beginning, without the need of textual instructions? Some musical toys have their sighted users accomplish very simple actions in a funny way, without the need of instructions for use. Moreover, video games show us that it is possible to separate the learning process of a complex task in small steps easy to master. We have made a game according to those principles and realized an experiment to test it. All the players managed to progress in the game but not all understood all the principles of the game. For this kind of game, we assume that players do not have to understand the game during the first contact but they have to be encouraged to continue interaction. At last, the increase of the difficulty level has to be very progressive.

Keywords: audio games, accessibility, usability test, interactivity, sound design, interactive music.

1 Introduction: Audio Games without Language for a Greater Accessibility

1.1 Without Visual Support, Languages Are Less Understandable

Accessibility in games is becoming every year a more important preoccupation in the industry and the report published by the IGDA marked a significant step [1]. The purpose of this study is to understand how visually impaired people can easily master new accessible audio games. Recent research and observation gave a good overview of the existing audio games [2], [3], [4], [5]. Actually, there are more than four

Z. Pan et al. (Eds.): Transactions on Edutainment II, LNCS 5660, pp. 176–186, 2009.

hundred games which are accessible to visually impaired players. It is possible to adapt mainstream games into accessible ones as it was done for quake adapted into audioquake [6]. It can be easier to think about the accessibility of a game starting from the beginning of its development. For this to be done, research proposed guidelines for a better accessibility in games [7], [8].

However, these studies do not consider accessibility problems related to language aspects: language is an inaccessibility factor for those who can't master it. Moreover, recourse to translation, which implies increasing development costs, doesn't allow us to target all the potential users: there is too much language, even among the most spoken, to operate all the translations. Now, an important part of existing audio games requires good understanding of the principles of interactivity: for this reason, the comfort given by language is tempting but even for those who master the adequate language, it can be an obstacle to amusement. Most of the actual and popular audio games need a lot of reading before players can start to play them efficiently. For players who don't master English, the game is not accessible. This is why it can be advantageous, for accessibility studies or for economical needs for international distribution, to develop games (followed maybe by other kinds of software) without communication via language.

1.2 Some Studies Encourage the Making of Audio Games

Are audio games without language realistic? A. Darvishi confirms that an information technology environment using various sonorities provides support for the understanding of the interactive process [9]. But this study doesn't say if sounds alone could be sufficient for correct interaction. For this reason again, the encounter of experimental research on audio games and the field of non linguistic communication could bring interesting results. This can be a way to orientate the purpose of audio games towards a more musical outcome: interactive music. Even without the help of tactile perception, J.L. Alty points out that music is usable as the main means of communication, with three centres of interest: the communication of musical algorithms, debugging, and communication for blind people [10]. Regarding this third area, A. Darvishi uses sound synthesis in a virtual environment accessible to the blind, each sound being the result of a particular configuration of the environment [9]. One of the problems of this approach could be the difficulty to convey precise information with sound synthesis, because of its very abstract nature, not really suitable for communication. However, there are various ways of arriving at a fuller level of communication. For example, B.N. Walker manages to communicate numerical data via a musical abacus [11]. In addition to scientific studies, other ways of investigation may be very helpful.

2 The Influences of Multimedia Experiments and Video Games

Audio games are not the only interactive enjoyment. We will now study how interactivity works with musical toys and what is the importance of language in them. They could be a good source of inspiration for working on the first contact between an audio game and its users. Then we will consider video games because some of them present interesting learning processes. Both multimedia experiments and video games could provide clues for the design of audio games without language.

2.1 Influences of Multimedia Experiments: Language Is Not Necessary When Simple Actions Must Be Done

Interactive artistic audio experiments provide clues for non linguistic communication. These multimedia experiments are often very abstract audiovisually, with no objectives and with no instructions for use. Users have to discover by themselves how to interact with sounds and their misunderstanding can be considered as a part of these artistic works.

Most of them are unfortunately inaccessible for blind people, due to the importance of the visual interface. This is even truer when these experiments show small interactive buttons on a large non interactive visual surface and when there is no musical information about what is under the cursor. In this case, information is given by language but often in a graphical, inaccessible way. This is the case for “Audiomeister”, an experimental musical mixer: it is simple, easy and funny to use but only for the sighted people. This first multimedia experiment suggests that immediate funny audio feedback is essential for the users. In “Snapper 8:8”, all the visual zones are interactive. It is thus possible for blind users to use the mouse and click randomly on the screen, but it must be on the zone covered by the experiment. For a sighed user, this musical toy doesn’t use instructions: users discover by themselves how to produce music, and it works well with sighted users. They realize two kinds of usual gestures: first, they move the mouse, anywhere on the surface of the experiment and then they click, nothing else. They just have to do these two usual gestures. We have presented these experimentations to some blind people; for them, it is harder: they need to know they must move the mouse and click. Moreover, for a blind user, the purpose of the experiment is not evident and they must be informed that it is a multimedia experiment. Other experiments are linked with movements of the mouse, without the need of clicking, for example to give an attractive musical rendering when the cursor meets specific fields. This kind of interactivity can be done without linguistic instructions for use.

The interaction in this type of experiment depends on exploration of zones. The “Spacializer” is a good example of this kind of musical toy. We can hear interactive sound, when the cursor is on specific zones, just by moving the mouse. This kind of experiment is even more easy to use because it needs the user to do not two but only one kind of gesture: to move the mouse.

However, these multimedia experiments present two problems for visually impaired people: the use of the mouse is not a habit, even if more and more audio games are based on this device and without visual perception, the purpose of these experiments is not evident.

Therefore, we suppose, on a strictly hypothetical way, that multimedia experiments using the keyboard may be more easy to use for blind people. Actual audio games use the keyboard. But this is not sufficient: keyboards have many keys and users must know which keys are useful.

Moreover, without the visual representation of the multimedia experiment, blind people can’t represent to themselves the purpose of this activity. “are there any objectives?” Or “what am I supposed to do?” Are things some of them said while they tried these experimentations. One particularity of these experiments is that they usually are toys where users don’t have to accomplish objectives. Concerning video games or audio games, they are no longer toys but games. We want to make a study on games rather than on multimedia experiments because we want to understand how users can elaborate a complex way of interaction with an application. Multimedia experiments only present simple ways of interaction with interesting audio feedback. There is no particular challenge to beat. Games, through different objectives, manage to complicate the interaction through a learning process. We think that the starting point of the learning process of audio games needs some characteristics of multimedia experiments in order to make a pleasant first contact. We have considered multimedia experiments and we will now consider video games to know more about the learning process.

2.2 Learning Process of Video Games: Simple Learning Processes May Be Combined for the Understanding of a More Complex Task

Over the last years, video games have been more often analyzed by scientific studies. These new studies allow us to define the nature of games, writing processes, technologies and the cultural impact [12].

It is easier for the player to learn a game with instructions included in the first step of the game. In this way, the player uses less his memory and he can practice without delay the instructions he learnt without risks of forgetting. Moreover, it is only when a lesson is understood by the player that the next instructions are given. A complex task may be divided in a great number of short or funny and easy to understand lessons often called "tutorials". For more complex games, we might fear that the amount of instructions is much greater. This will be more often not true than true. Linguistic communications are sometimes used but not always. The first basic actions are explained with verbal instructions and the player must understand by himself the combination he can make.

The great difference with simpler games is that learning is no longer presented before the game but incorporated in the game itself. Tutorials often have the following characteristics:

- No or very few « game over » situations;
- Players face situations that can only be resolved in one way;
- Clues make the resolution easy. Clues may be audio, visual and / or tactile;
- There are few advantages to the player to go back;
- Players should be very interested in going further.

Approaching each of these characteristics from an auditory rather than a visual point of view may be of great interest. So, there is a paradox. The number of instructions does not depend on the degree of complexity of the game. For complex games, developers are looking for other ways to make players learn and quickly enjoy themselves without being discouraged. Considering, on one hand, that perhaps simple interactions don't need instructions, as suggest to us multimedia experiments, and on the other hand, that the understanding of simple interactivity may be combined, it should be possible to develop audio games without linguistic communication.

2.3 Playing Toy-Games

For an audio game without language to work, the game should also be a toy. The learning process isn't homogeneous. It depends on each user individually and in order to avoid discouraging them, each action should bring enjoyment. This was one of the objectives of a particular game: the phase project [13].

Phase is a game which can be understood in three different ways: in a visual, an audio or a tactile way. Area exploration and gesture have both been integrated, each of these kinds of interactivity provides musical transformations in the game. For this reason, phase is a toy because each action has an amusing outcome. In the same way of the musical experiment we have considered, players can produce music just by moving a haptic joystick. They can produce other sorts of spatialized sound by exploring zones. Phase is also a game and more exactly a race game where players can catch a musical entity running along the horizon of an audio tactile landscape. The rule of the game is also induced by the musical outcome. The faster the players go, the more the music is exciting and interesting. Because of the constraints arising from the context of an exhibition, this game was voluntarily very simple. Phase was used by people of all nationalities, without having to be given instructions, except in a few cases. A lot of blind people successfully played this game as a toy and some of them succeeded in the game. These are very encouraging results for using this type of design in audio games. We have made a game on similar principles, for a more standard configuration: no more tactile feedbacks, but the use of the keyboard, and the audio feedback in stereo only.

3 Experiment

3.1 General Hypothesis

Ideally, a game should be playable as soon as a player has a first contact with it, during the learning phase and without the help of someone else.

3.2 Study Environment

During tests of preliminary projects without verbal instruction, we have noticed that it is better if the player chooses himself the adequate moment for interaction. The making of an action game where timing is important seems harder to do.

This is the reason why we have made two maze games. Players have all the time they want to interact without any consideration of timing. The first one, named "Pyvox, musical maze" uses an included verbal tutorial but it was also tested without these instructions. Results were encouraging: in the non-verbal version, the players did not manage to progress as far as in the verbal one, but almost all the players managed to play the game [14]. The second one, named logically "Pyvox 2, more musical mazes" doesn't use verbal instructions. The beginning of this game also works as a musical toy. For both of these projects, we have used usability tests as often as possible since the beginning of the development. During these tests, we don't present any instructions to the player except that "it is a game and for the purpose of the test, it is better if nothing is told about the rules of this game." Then we consider the progression, the keys the player uses and the problems of understanding he (or she) meets. After each test, we include new modifications to the game.

After this iterative process, we obtained a game with the following features: the player directs a character in a seventy floor labyrinthine tower, seventy floors corresponding to seventy game levels which can be explored one after another in an unchanging order. The character can also be considered as a cursor that can be moved on a grid from one square to the other. The player can move the cursor towards target areas but obstacles block up some access paths. This game is a maze with a square-to-square moving system, divided into game levels, each level presenting an exit and a certain number of walls. The aim of the game is to teach the player to recognize an exit sound just by using non-speech audio. The game also aims at making the player recognize the sounds coming out of the walls in order to avoid them without hitting them; this recognition is obtained with similar non-verbal principles: the walls emit sounds in order to make the player feel where the obstacles lie. This principle is conveyed implicitly: we wanted the sounds to be slightly unpleasant so as to force the player to take his character away from the walls. The link between those sounds and the walls is understood rather quickly. More pleasant sounds can then be heard. As a first contact, the game introduces the character sleeping. The keys of the keyboard almost all trigger alarm sounds. The closer the player gets to the arrow keys, the louder the alarm sounds become. The use of the arrow keys wakes the character up and the exploration of the maze can start.

3.3 Aim of the Study

We want to optimize the handling of audio games by tackling the problems linked to language. We also want the visually impaired player to be able to learn the rules of the games without verbal instruction.

3.4 Studied Population

The testers selected for the experiment are two groups of teenagers who have participated in the international computer camp 2007 (ICC) in the Arla institute from Espoo, Finland. They are all visually impaired. During the first week of the ICC, the teenagers were between fourteen and seventeen years old. During the second week, the first group left and older teenagers aged eighteen to twenty two came instead. The experiment was realized during specific workshops about audio games, so the testers had very little time - about three hours - to try more than a dozen games. Some of the testers wanted to put an end to the tested game in order to try the other ones before the end of the workshop. The other presented games were: “Terraformers”, “Shade of Doom”, “Sonic Zoom”, “Sarah in the Castle of Witchcraft and Wizardry”, “Super Egg Hunt”, “Mudsplat”, “Super Liam”, “Pyvox Musical Maze”, “Tampokme”, “Chrono Mouse”, “Descent into Madness” and “Top Speed 2”. However, all these games were not always presented at the same time due to computer reconfiguration. Due to the conditions of the workshop, it was not possible to make a preliminary study to know for example if each tester was familiarized with audio games. The players were here to discover audio games and they had very little time; also, during the tests, we had to help players who were trying other games. Unlike with Pyvox 2, we had to explain the rules of those other audio games since they were not meant to be played without prior knowledge of instructions in the manual.

3.5 Variable Independent from the User

We counted the duration of each game and the highest level reached.

3.6 Variable Dependent from the User

We tested a unique version of our game, but we consider separately the two groups of testers, with different ages:

C1: testers from the first week of the ICC07, between fourteen and seventeen years old.

Unfortunately, we only managed to gather seven testers during the first week. Workshops did not occur all the days, and the first attempts were lost due to the preparation of the audio games for the ICC.

C2: testers from the second week of the ICC07, between eighteen and twenty two years old.

We managed to collect sixteen testers during this second week.

3.7 Instructions

In this study, the testers are not faced with the device freely, they are aware of the following details: they are going to try a game, the purpose of which is not given. They can play as much as they want. They can give up the game whenever they want to but they cannot resume it. They can adjust the sound volume with the controls indicated by the experimenter. The controls to adjust the sound volume are tested by the players. The experimenter starts the game and leaves the players to use the device until they want to stop the game by themselves.

Each tester only does a single game without preliminary training.

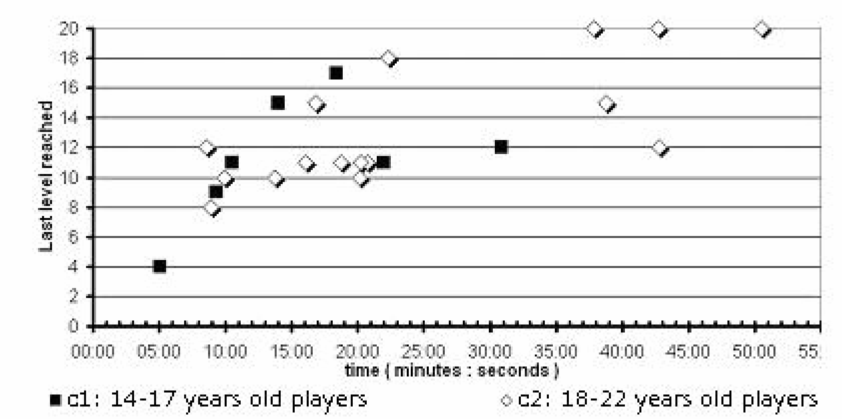
3.8 Operational Hypothesis

We assume that our game without language but including the learning principles previously suggested could be understood and properly played by players from the two age groups. However, we have noticed from previous tests that there could be very different reactions between players, depending on their ages. Adults without game experience for example may have great difficulty to handle audio games and to understand them. For this reason, we want to compare these two groups, without manipulating any other variables in order to have a better understanding of the different kinds of reactions to a same material.

4 Results

All the testers managed to pass the three first levels. Six testers ended the game at the sixth level. The majority of the testers, sixteen players out of twenty three, have played the game between eight and twenty four minutes. Six players have played more than thirty minutes. The five more persevering players are from the older group and the most persevering has played fifty minutes. The four players who have reached the highest levels are all from the older group too.

Table 1. Main results of the two groups of players



5 Discussion

All the testers managed to pass the three first levels (Fig 1). However, it doesn't mean that all the testers understand the principles of the game. At this point, the majority of the testers did not represent to themselves the game as a maze game. We have noticed during the ICC, that some sighted staff members failed to pass this third level, staying on the left of this audio maze without being able to go to the right side. For this reason, we consider that the level design of the first level should be improved with a lower difficulty level.

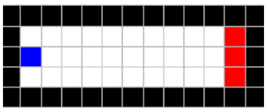


Fig. 1. The third level of Pyvox 3. The starting point is on the left and exits are on the right.

The increase of the difficulty level seems particularly problematic for the level number eleven (Fig 2.). Six testers ended the game on this level. Six others did not manage to reach this level and ended the game before. Eleven testers managed to pass this level.

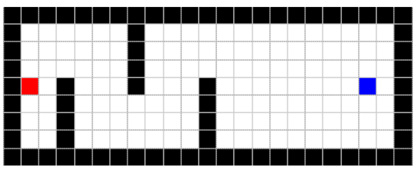


Fig. 2. Level eleven of Pyvox 2. The starting point is on the left and the exit is on the right. The player has to recognize and avoid three walls on the path before he can find the exit.

We think that the testers who managed to pass the level number eleven have understood the main principles of the maze game: players have to find an exit and the exit is indicated by a specific audio rendering using stereo and pitch variation depending on the position of the player's character compared to the position of the exit. However, at this point, we are not sure whether the players know how to identify walls without hitting them and recognizing the related sounds on impact. The maze is still relatively simple.

We have six testers who have played more than half an hour. The five more perseverant testers are from the older group of testers. The four better players who have reached the higher levels are also from the older group.

Three testers of the older group managed to pass the level number eighteen (Fig. 3). For these players, all the rules of the game seem to have been well understood.

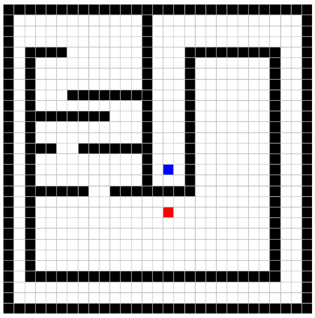


Fig. 3. Level eighteen of Pyvox 2. The starting point is in the center. The exit is near below. The player's character has to take the path on the right and go round the maze.

Despite these observations, there does not seem to be strong difference between the two groups.

We assume that not all the players understood all the rules of the game, but these results are related to their first attempt, without any preliminary training. Other attempts from the players should allow them to progress further in the game.

We consider that the players do not have to understand the game but have to experience enjoyment during the progression. What is important in games is the feeling of enjoyment. This particular sensation in games has been well studied since the works of Csikszentmihalyi about the flow theory [15]: a game is good enough if it strikes a balance between the challenge proposed by the game and the player's abilities. The flow zone, which is the feeling produced by good games, is in the middle of two opposite sensations: anxiety and boredom. It is not directly linked with the concept of understanding. Jenova has worked on the flow theory and used it to give us a very interesting game named "flow", without any verbal explanation [16]. Pyvox 2 is a game, and we consider that the difficulty to understand the principles of the game can be a part of the proposed challenge. However, the learning process could be facilitated and reviewing the level design and the sound design should help to do so.

6 Conclusion

We have envisaged that multimedia experiments allow the players to play with music as if they were playing with a toy: there is no objective. Then, we have seen that there are many ways to interact with these toys: by clicking or just by moving the mouse. Some of these experiments don't need instructions for use for sighted players who understand by themselves how to play. There are no multimedia experiments which are specifically adapted for blind people, we suppose that three factors are important for a good use of these multimedia experiments: usual gestures, funny and immediate audio feedback and finally, a good representation of the purpose of the multimedia experiment. In the next part, we have considered video games and their learning process: a complex task may be divided in many simple lessons that are easy to master, all included as a first part of the game. The first contact between players and audio games without language or other accessible audio software should combine the qualities of toys and games. Toys are fun to manipulate without instructions. Games encourage progression in the manipulation by a pleasant learning process. An audio game without language could imply that first, the player discovers a toy and then, because the audio rewards are not all the same for all the interactions, he is looking for the manipulations which provide the best encouragements. So the sound toy should successfully become an audio game without language. The making of an audio game without language implies that we can't develop a specific type of game without making other games for prerequisite knowledge: how to use the keyboard, which keys are really useful and which keys are not, what are the consequences of their usage. This is a whole chain of minigames that should be linked up in order to understand and play the desired game.

We have tried to implement those principles in a maze game. We are satisfied with the results, considering that they concern the first attempt without any preliminary training and without any explanation. However, we think that the learning process can be facilitated by making a level design with a more progressive increase of the difficulty and by improving the sound design again.

We are currently working on a third audio game named "Pyvox 3", it is also a maze game without verbal instruction and with a multiplayer feature. Thus, we think it can be easier for players to communicate hypotheses about the rules of the games.

Acknowledgements. This work has partially been realized with the support of CECIAA, in the framework of a CIFRE contract.

Our thanks to the ICC organization and to the Arla institute of Espoo, Finland, for giving us the possibility to make this study.

References

1. IGDA: I. G. D. A.. Accessibility in games: Motivations and approaches (2004), `http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf`
2. Archambault, D., Ossmann, R., Gaudy, T., Miesenberger, K.: Computer games and visually impaired people. Upgrade viii/2 (July 2007)

3. Feir, M.: A decade's accessible gaming: an old editor's anniversary observations. Audyssey 49 (2006)
4. France, M.: Audio game survey results. Audyssey 50 (2007)
5. Gaudy, T., Natkin, S., Archambault, D.: Classification des jeux sonores selon leur type de jouabilité. In: Proceedings of Handicap 2006 conference (2006)
6. Atkinson, M.T., Gucukoglu, S., Machin, C.H.C., Lawrence, A.E.: Making the mainstream accessible: What's in a game? In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 380–387. Springer, Heidelberg (2006)
7. Ossmann, R., Miesenberger, K.: Guidelines for the development of accessible computer games. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 403–406. Springer, Heidelberg (2006)
8. Grammenos, D., Savidis, A., Stephanidis, C.: Unified design of universally accessible games. In: Stephanidis, C. (ed.) HCI 2007. LNCS, vol. 4556, pp. 607–616. Springer, Heidelberg (2007)
9. Darvishi, A., Guggiana, V., Munteanu, E., Shauer, H.: Synthesizing non-speech sound to support blind and visually impaired computer users, computers for handicapped persons (1994)
10. Alty, J.L., Rigas, D., Vickers, P.: Sing music as a communication medium (1996)
11. Walker, B.N., Lindsay, J., Godfrey, J.: The audio abacus: representing numerical values with nonspeech sound for the visually impaired (2002)
12. Natkin, S.: Video games and interactive media: a glimpse at new digital entertainment, ak peters (2006)
13. Rodet, X., Lambert, J.P., Cahen, R., Gaudy, T., Guedy, F., Gosselin, F., Mobuchon, P.: Study of haptic and visual interaction for sound and music control in the phase project. In: Proceedings of the 2005 conference on new interfaces for musical expression, Vancourer, May 2005, pp. 109–114 (2005)
14. Gaudy, T., Natkin, S., Archambault, D.: Playing audiogames without instructions for uses: to do without instruction leaflet or without language itself? In: Proceedings of CGAMES 2006, Dublin, November 21-24 (2006)
15. Csikszentmihalyi, M.: Flow: the psychology of optimal experience (1990)
16. Jenova, C.: Flow in games, mfa thesis, university of southern California (2001)

4.4 [AAATE'2003]

Titre	**Blindstation : a Game platform adapted to visually impaired children**
Auteur(s)	Sébastien Sablé[(a)] et Dominique Archambault
Affiliations	[(a)]Université Pierre et Marie Curie-Paris 6
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Assistive Technology – Shaping the future, Proceedings of the AAATE'03 Conference
Editeur(s)	Ger Craddock, Lisa McCormack, Richard Reilly et Harry Knops
Lieu/Date	2003
Pages	232–236

Voir mémoire section **3.4.1**, page 40

Blindstation : a Game Platform Adapted to Visually Impaired Children

Sébastien Sablé and Dominique Archambault
INSERM U483 / INOVA - Université Pierre et Marie Curie
9, quai Saint Bernard, 75,252 Paris cedex 05, France
Sebastien.Sable@snv.jussieu.fr

***Keywords**: visually impaired, multimedia, multi modality*

Abstract: The TiM project intends to develop and to adapt computer games for visually impaired children. A game platform, the blindstation, was developed to adapt existing content or create some new games. It provides a set of Python functions to describe those games in an abstract way, independent from their representation. The platform can then render the game in a multi-modal way using the screen, keyboard, mouse and joystick, but also using some specific devices like a Braille terminal, 3D sound, a tactile board or a speech synthesizer. The rendering is done according to an XML style sheet which describes the available resources. It can be customized depending on the available devices but also on the user's choices and disabilities. Several games have already been developed in different types (action, adventure, exploration...) and are currently tested in schools specialized in visually impaired children.

1. The TiM Project

The game engine presented in this paper is part of the TiM (Tactile Interactive Multimedia) project [TiM2000]. The purpose of this project is to offer multimedia computer games intended for young children who are blind or severely visually impaired. TiM is also addressing visually impaired children with additional impairments in form of slight to moderate degree of learning or cognitive disability and/or a physical impairment [TiM2001]. These games are planned to be used by the children in an autonomous way, without assistance of a sighted person.

To reach the needs of these children, the games have to be adaptable to their specific needs and to all the specific devices they use, each corresponding to a specific modality: tactile boards, Braille displays, speech synthesizers, as well as standard devices like keyboards and sound devices or adjustable screen settings (scalable font prints, customizable colors and contrast...).

Additionally, the specificity of modalities used by visually impaired children often makes it necessary to modify the scenario of interaction. For instance, a lot of games are based upon the global vision of the layout and the visual memory.

TiM plans to handle all the aspects necessary to ensure the development of high quality games by developing tools that allow the adaptation of existing commercial computer games ensuring that educational and play contents are incorporated; It has also developed specific contents meeting the particular needs of a young blind or partially sighted child. TiM will also include complete material about adaptability, methodology and guidelines.

2. Blindstation overview

2.1. Game Engine Principle

In order to simplify game creation and adaptation in the TiM project, a game engine or platform, named Blindstation, was developed. To easily understand the concept of a platform, people could think about the existing hardware game platforms, like Nintendo or PlayStation.

While the TiM platform is purely made of software and runs on a standard PC, it has the same features: it provides to developers some high level functionalities to interpret a TiM game script with its resources, to process the data and to drive all the devices that are connected to the player's computer. Thus simplifying game design.

2.2. Adaptability Specificities

The mainstream computer games are usually designed especially to be used through a standard multi modal interface (graphical display, mouse and speakers). So the approach for games development is based on visual conception: interaction objects of the game are generally represented in the design software as pictures which are attached to some parts of the screen.

In order to be able to design games that are independent from the modalities and can use specific devices, the approach chosen by the TiM project is to use a modality-independent model [Arch2001]. This is the reason why in the TiM platform, the game are described in 2 distinct parts: On the first hand the game scenario is written in a script using some abstract high level components. On the other hand, the resources (audio samples, music, texts, pictures, animations...) that can be used to render the different components are specified in a structured style sheet.

It is only during the execution of the game, that the platform, using the script and the style sheet will render the interface in a multi modal way according to the environment and the needs of the player.

3. Game Script

```
from blindstation import *
class Intro(Scene):
    def __init__(self, game):
        Scene.__init__(self, game, "test_scene")
        self.label = Widget(self, "label")
        self.button = Widget(self, "button")
        self.button.validate_callback = self.callback1
        self.listener = Listener()
        self.source = Source("source")
        self.source.source_attach = self.source_callback
        self.source.queue("thunder")
        self.source.queue("bird")
        self.source.play()

    def callback1(self, name):
        print "callback at validation of button Widget"
        self.next = Menu

    def source_callback(self, name):
        print "callback at the end of the sound"
```

A simple example of a script using the Blindstation

The scenario of a TiM game scenario is described in a game script written in the Python [Python] computer high level language and using the components provided by the Blindstation. This script is interpreted by the Python interpreter and uses functionalities of the platform through the "platform API". An API is an abbreviation of Application Programming Interface, a set of routines, protocols, and tools for building software applications.

As described above, the API is organized in different components completely independent of the game representation and providing functionalities to the game designer in order to simplify game design.

3.1. Standard Components

Some components are used to describe the game flow and can be found in most traditionnal game platform. For example, the Scene, just like in movies, is a way to divide a game in many small coherent sequences; the Game component represents the context in which the whole game is running and is responsible for initializing many low level aspects of the platform (like loading the many libraries, initializing the display, the sound, the braille display...) and dealing with scenes changes.

The interaction in the platform depends mainly on events being generated. Each time something happen in the game (for example when a user interacts with a component), an event is emitted. Generally events trigger some callbacks defined in the game script. Some simple components like the Timer make it possible to trigger some events at some given time.

3.2. User Interaction Components

Those components are used to interact with the player. Since the interface of the game has to be adaptable, those components are very versatile. For instance, there is a Widget component. Depending on the context, this component can be rendered on the screen at a specific position with some text and pictures, but it can also be accessed in Braille from a Braille device or read thanks to a speech synthesis or activated with a tactile board cell.

Other components include a Navigator which makes it possible to "navigate" in a virtual environment with a joystick for example, a Listener indicating the player position or spacialized sounds.

Components are generally associated to a resource name that will be used by the platform to recognize which resources are available in order to render this component.

3.3. Resources Style Sheet

```
<blindstation>
<scene name='test_scene'>
   <color r='255' g='255' b='255' a='255'/>
[...]
   <widget name='button'>
      <text lang='us'>Validate for Menu</text>
      <text lang='fr'>Valider pour le Menu</text>
      <zone x='50' y='50' l='500' h='50'/>
      <shortcut key='SPACE'/>
      <color r='0' v='255' b='0' a='128'/>
   </widget>
   <sound name='thunder'>
      <file lang='fr' name="data/thunder.wav"/>
      <file lang='en' name="data/thunder.wav"/>
   </sound>
[...]
</scene>
</blindstation>
```

The Style Sheet corresponding to the previous script

As seen before, the scenario of TiM games is described in a script that is completely independent of the final representation. Since the games have to be adapted to different multi modalities, the resources available to represent a given component (like sounds, text, pictures) have to be stored in a structured way. An Extensible Mark-up Language (XML) document called the Style Sheet is used.

This idea is based on the same principle that can be found in web pages. In web documents, the structure of the text is described in an HTML file which contains only the structure of the text (a title, a section, an item of a list). Then another file called Cascading Style Sheets (CSS) is responsible for the final aspect of the document [WAI].Our style sheet also contains information in order to deal with many languages.

4. Platform internals

The Python language is particularly well suited for the needs of this project since it is high level and interpreted. However some power consuming parts were optimized by using C code wrapped using SWIG [SWIG].

The platform is light enough to run on a multimedia computer with a low configuration, a sound card, loudspeakers and a CD-ROM drive. The platform is portable and can run on systems with Windows 95 and upper or with GNU/Linux and possibly MacOS.

4.1. Game flow

At initialization, the platform will detect, if technically possible, which devices are available on the computer and initialize them. Devices can also be specified in a configuration file.

Then the platform has to deal with scene changes and to listen to all the possible events. An event is an information that propagates itself to trigger some functionalities of components and is generally generated when a user does an action. Depending on the nature of the event, it can be treated internally for performance issues.

4.2. Resources management

One of the operations that consumes a lot of memory is to load pictures, sounds, video, fonts and other resources. The platform uses a cache mechanism to load resources only when needed and to ensure that only one instance is loaded at any time. In order to guaranty good performances while playing, the platform also does use streaming when necessary, for example when playing big sound files.

4.3. Adaptative representation

The Blindstation is responsible for the rendering of the components described in the game script using the resources style sheet. Many parameters are responsible for the final representation: available devices, user profile, data collected about the user during the game, user's preferences stored in a configuration file... On top of that, multilingual features have been integrated so that the same game should be played by people speaking different languages. Using all those pieces of information, the platform is able to select the best possible representation for the user interface.

4.4. External libraries

To improve extendibility, the project uses some external libraries for specific things: Libbraille [Libbraille], which was created for the TiM project, is responsible for all the low level interactions with braille displays. Libspeech [Libspeech] drives the speech synthesis while libboard deals with tactile boards.

Other libraries developed externally are used, like OpenAL [OpenAL] for interactive, primarily spatialized audio. It can play the sounds in the player's headphones or a 5.1 surround system of speakers. Finally SDL [SDL], a cross-platform multimedia library, is used to provide fast access to the graphic frame-buffer, audio devices, keyboards and joysticks.

5. Current Status And Further Works

5.1. Adapted games

The first version of the Blindstation was developed and several games have been designed and are currently evaluated with children :

- *Reader rabbit's: Toddler* is an adaptation of a mainstream discovery game designed for very young children. It features many small puzzles based upon sound and tactile recognition, for children who cannot read
- *MudSplat* is an arcade like game where the player defeats mud throwing monsters by squirting water at them. It is mostly based on sound and navigation systems.
- *Tim's journey* is an exploration game. The player solves a quest, by exploring in real

time a surround sound environment and listening to stories

- *X-tune* is a musical construction game where the player can sequence and play with different sound environments
- *Magic dictation* is an educational game for learning reading and writing; it is particularly well suited to children learning Braille

5.2. Game builder

A TiM authoring software is planned that will facilitate the adaptation or development of games by providing a friendly drag and drop interface to completely create the game. This tool is destined to be used by non-computer literates: professionals working in special schools or rehabilitation centers, resource centers, educators, teachers, parents…

5.3. Evaluation

The game are evaluated in different sites across 3 countries including special schools for blind children, rehabilitation centers for multi-handicapped blind children, ordinary schools which receive blind children and a parents association. The studies are carried out by a team of specialists from various disciplines: educators, teachers, ergonoms, psychologists and therapists. The feedback to the software developers and game content designers will improve the games and the representation of components in the platform.

Acknowledgements

The TiM project is funded by the European Commission, on the program IST 2000 (FP5 - IST - Systems and Services for the Citizen/Persons with special needs) under the reference IST-2000-25298. The contents of this paper is the sole responsibility of the authors and in no way represents the views of the European Commission or its services.

References

[TiM2000]: Archambault, D. and al, TIM : Tactile Interactive Multimedia computer games for visually impaired children., , http://www.snv.jussieu.fr/inova/tim

[TiM2001]: Archambault D., Burger D., and Sablé S., The TiM Project: Tactile Interactive Multimedia computer games for blind and visually impaired children, in Assistive Technology -- Added Value to the Quality of Life, In: Marincek C., Bühler C., Knops H., and Andrich R. (eds.), Proceedings of the AAATE'01 Conference ,2001

[Arch2001]: Archambault, D. and Burger, D, From Multimodality to Multimodalities: the need for independent models., in Constantine Stephanidis (ed.) Proceedings of the UAHCI'01 conference - Universal Access in Human-Computer Interaction (joint with 9th International Conference on Human-Computer Interaction) – Towards an Information Society for All. ,2001

[Python]: , Python, an interpreted, interactive, object-oriented programming language, , http://python.org

[WAI]: Chisholm, W., Vanderheiden, G. and Jacobs, Web Content Accessibility Guideline 1.0, Web Accessibility Initiative, , http://www.w3.org/TR/WCAG10/

[SWIG]: , Simplified Wrapper And Interface Generator, , http://www.swig.org

[Libbraille]: Sablé S. and Archambault D., Libbraille: a Portable Library to Easily Access Braille Displays, in Proceedings of the 8th International Conference ICCHP 2002 - Computer Helping People with Special Needs ,2002

[Libspeech]: , , , http://libspeech.sourceforge.net

[OpenAL]: , Open Audio Library, , http://www.openal.org

[SDL]: , Simple Direct Layer, , http://libsdl.org

4.5 [Technology and Disability, 2006]

Titre	**A simple game generator for creating audio/tactile games**
Auteur(s)	Arnaud Puret[(a)], Dominique Archambault, Mohamed Slimane[(a)] et Nicolas Monmarché[(a)]
Affiliations	[(a)]Polytech'Tours
Type	Revue Internationale, avec Comité de lecture
Publication	Technology and Disability, IOS Press, Amsterdam, The Netherlands
Editeur(s)	Christian Berger-Vachon
Volume/Date	Vol. 18 (4), 2006
Pages	227–236

Voir mémoire section **3.4.1**, page 40

Technology and Disability 18 (2006) 227-236
IOS Press

A simple game generator for creating audio/tactile games

Arnaud Puret [a], Dominique Archambault [b,*], Nicolas Monmarché [a], and Mohamed Slimane [a]
[a] *Université François-Rabelais de Tours, Laboratoire d'Informatique, Polytech'Tours, France*
[b] *University Pierre et Marie Curie, Paris, France*

Abstract. We present in this paper a new game generator that aims at facilitating educators to provide new games scenarios to young visually disabled children. A game scenario description is stored in a XML file without any necessary knowledge of this format from the educator. Moreover, scenarios could be easily shared via a web server.
Keywords: Tactile interface, computer game, game authoring tool, blind, visually disabled.

1. Introduction

findit! is one of the games developed in the framework of the *TiM* project [1] (Tactile Interactive Multimedia computer games for visually disabled children). It is a very simple Tactile/Audio discovery game, intended for very young children or children with additional disabilities. The player must associate sounds with pictures on the screen or tactile information on a tactile board. Recorded comments and clues are associated with the items.

This game is very appreciated by the children for whom it is intended, but its potential resides in the variety of contents that can be produced for it. To facilitate the design of new contents for this game a game generator, with an intuitive drag and drop interface, was developed.

In this paper, we will first give a quick overview of the *TiM* project, focusing on the game development, a description of *findit!* itself, and then present the game generator.

2. The *TiM* project

The overall aim of the *TiM* project was to provide young visually disabled children, with or without additional disabilities, with multimedia computer games they can access independently, that is without any assistance of a sighted person.

Original studies about adaptation of game interaction situations to non visual interfaces were performed [4], and a development platform was developed, called *"blindstation"*, facilitating the design of games intended to visually disabled children [6]. A set of games corresponding to various interaction situations was designed and these games have been evaluated with visually disabled children.

The *blindstation* is an API[1] using a modality-independent model [2]: one of its most important features is the separation between the game scenario and the multimodal and multilingual resources:

– Game scenarios are implemented independently of the devices and of the language. A game scenario is a Python script[2] that uses the *TiM* Platform API components. Those components are able

*Correspondence to: Dominique Archambault, University Pierre et Marie Curie, Inserm U483/Inova, 9 quai Saint Bernard, 75252 Paris Cedex 5. E-mail: dominique.archambault@upmc.fr

[1]The Application Program Interface is a set of functions for building software applications
[2]The Python language: http://www.python.org

ISSN 1055-4181/06/$17.00

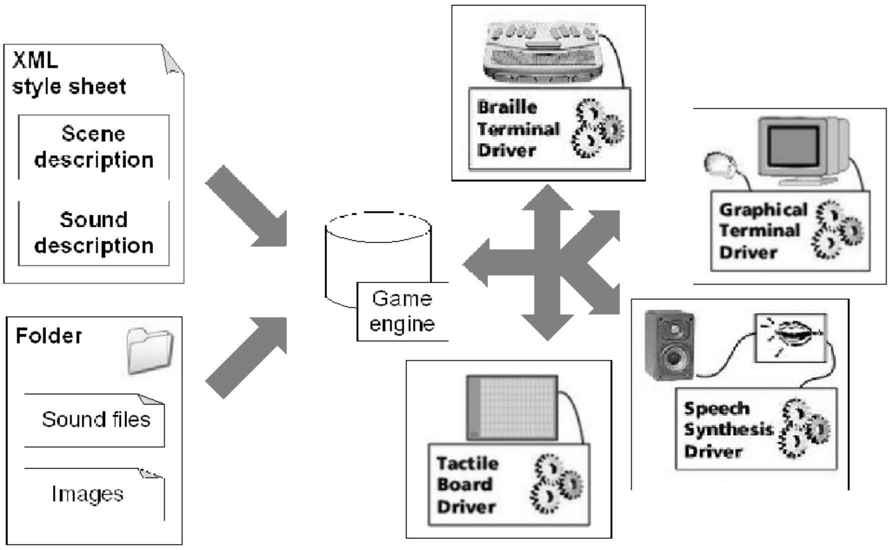


Fig. 1. The *blindstation* based game principle

to manage different kinds of special devices corresponding to various modalities (tactile, audio, large screen displays).

– Each game scenario is associated to a style sheet that contains all the necessary resources that are needed to render each component using each modality, and when it is relevant in each language (see figure 1). This style sheet is an XML document with a specific DTD[3] that was designed to fit our needs.
– Texts are embedded inside the style sheet, the other media files (sounds and picture files) are stored into the game directory. All resources that are specific to a language (text, recorded voices, graphics containing texts) are associated to this language in the XML style sheet, using a "lang" attribute. In that case, it is obviously necessary to provide multiple version of any relevant sound or graphical file.

3. *findit!*

A few accessible games have been created for people with visual disabilities: board games like UA-Chess [3], 3D games with Terraformers [8]... The main aim of these games is not learning but entertainment. However they are all based on the use of sound to interact with the player. Research works in haptics [5,7] have shown that the tactile sense has a great potential to help blind and partially sighted people to understand their environment.

Following these points, the game *findit!* was created as a tactile/audio association game for young blind or partially sighted children. The aim of the game is to make the player associate a sound with its tactile or visual representation. The game necessitates a tactile board with a tactile overlay (see figure 2), and a sound card with a speaker. All information presented on the tactile board is also displayed on the screen.

findit! is intended to work with various content sets, called templates. A first content template was developed during the *TiM* project. It is based on 4 cats in 4 different situations: the purring cat, the kitten, the angry cat and the hungry cat (see figure 2). This game contains 4 illustrations, 4 sounds, 4 recorded audio comments (like *"This one is pleased, this must be Happy, the purring cat"*), etc. A layout for tactile board was developed with the cat represented in the same way in each case (a round piece of fur):

– on a pillow for the purring cat,
– with corners in sand paper for the angry one,
– close to a plastic plate for the hungry one,
– the kitten is smaller and in a small basket.

The game starts in a discovery mode, in which the child is asked to press each of the items on the tactile board, and then he/she will hear the associated sound, and the corresponding recorded comment.

When all the available items have been discovered, the game goes to the Association mode where the child

[3]The Document Type Definition is a grammar used to validate XML files

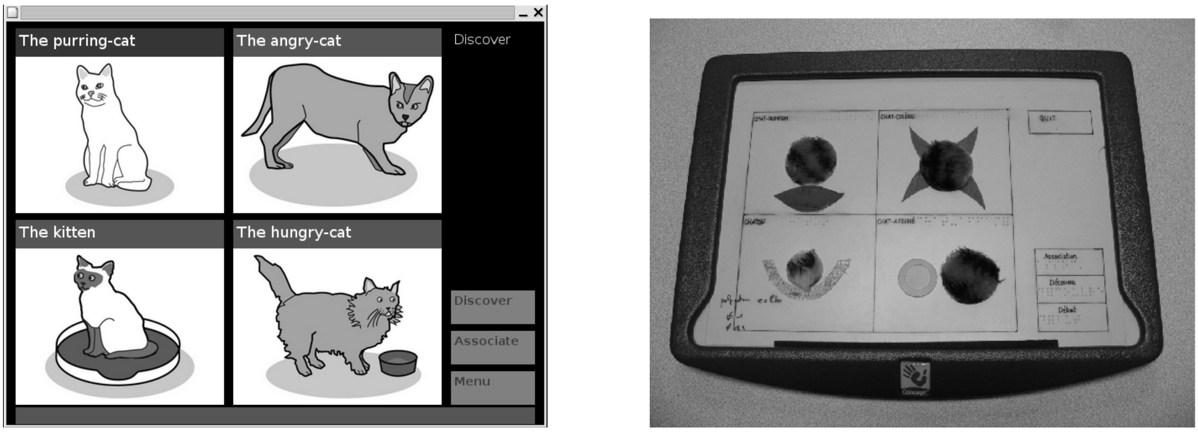


Fig. 2. *findit!*: The graphical interface (left) and the tactile board (right).

hears one of the sounds and then has to find out which one it is. The answer is given on the tactile board by pressing the corresponding shape or on the computer keyboard. If he/she fails, a recorded clue is played, referring to the sound and to the tactile figure (for instance: *"Listen, he is purring. Can you find him? He is very sweet!"*).

4. The game generator

As said above, the interest of this game resides in the variety of content templates that can be proposed. Additionally it would be very valuable that the educators themselves can produce their own templates and eventually share them on a web server. That is why a game generator was developed.

4.1. Presentation

The generator was designed in a very flexible way. It is based upon various configuration files, which facilitate its adaptation to other kinds of games. The configuration files are of 4 types (see figure 3):

1. the XML game template: this is just a minimalist XML game file. It gives the right position, size and color to the screen vignette elements. Neither sound file nor picture is used in a template, every location for them are free. Moreover, the template can be an old game (with sound and picture files) loaded into the generator to provide a complete example or to be modified.
2. the XML DTD: this file is used to validate a template or a game file. The DTD contains the right syntax for these XML files.
3. the generator front-end modules: these are the active parts of the generator. A module is a Python script which collects the game properties, builds the associated generator property panel, displays it and provides the tools to modify the game information.
4. the XML conversion file links the game properties with the generator properties. The structure of this file allows to create different types of game, it describes the location of the vignettes and all their properties inside the game file. In the same way, the background elements and the menu configuration are taken out from the game file.

At the end of the game creation, the generator produces a new XML file and the directory with the needed sounds and pictures.

The XML conversion file indicates to the front-end modules where to get the right properties, how to display them and what tools can be used. The game properties are not displayed directly, the generator gives the user a synthetic view of the game thanks to the XML conversion file: a front-end module creates a user friendly property panel.

The user can only see the game template and the new game files. All the other files are created once and for all by the game creator. These files (the basic template, the DTD, the XML conversion file and the front-end modules) must be shared to make possible the creation of new games by many users (by sighted persons).

4.2. The generator technologies

The game generator was developed using the Python scripting language. this is the language used in the TiM project (the game *findit!* was written with this language). The Python language is (semi) interpreted, this

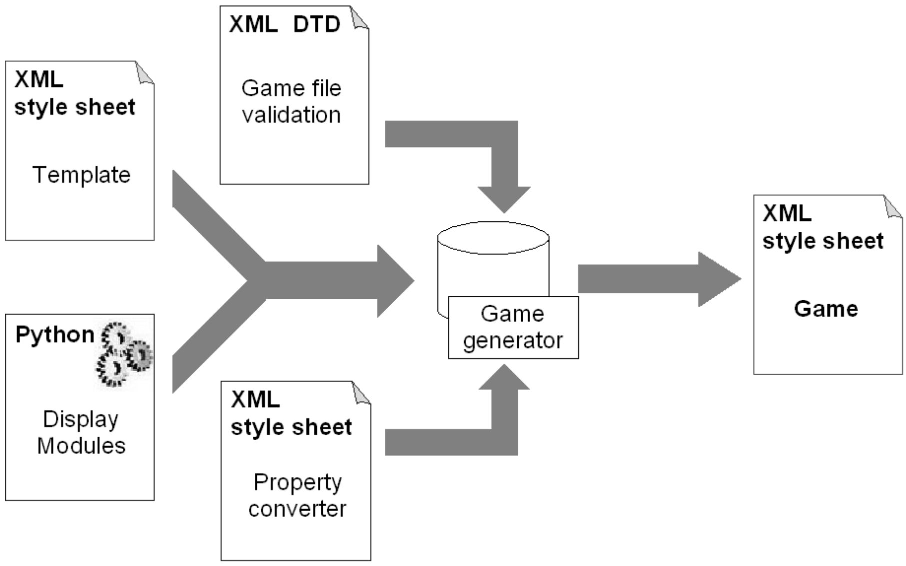


Fig. 3. The generator configuration files

feature is very useful to create new front-end modules and add them dynamically to a running software:

- the generator reads the XML configuration file and creates the graphical interface according to its content, the modules are dynamically loaded.
- the number of items present in the game is not fixed, the user can add and delete one of them, the corresponding front-end modules are loaded and unloaded accordingly.

The graphical display uses wxPython[4], this is the Python version of the WxWidgets[5] GUI API. It is very efficient and well documented.

The audio library is the Snack Sound Toolkit[6]. It is built for Python and Tcl scripting languages and provides high level functions (the link with the operating system is managed by the library). Snack provides directly multiple sound displays like sound wave and spectrum displays.

All these products are free and open-source. The Python, WxWigets and TkSnack combination allows the creation of a portable software across operating systems (at least GNU/Linux and Microsoft Windows). Thus a majority of users will be able to use the game generator without having to adapt the program.

[4]wxPython: http://www.wxpython.org
[5]wxWidgets: http://www.wxwidgets.org
[6]snack: http://www.speech.kth.se/snack

4.3. *The main window*

Now, let us have a look at the generator's main window. It is separated into two parts:

1. the left part represents the game structure, it is displayed like a tree to be extensible and its content is described in the XML conversion file. The tree is automatically generated according to the number of items present in the game and its nodes are the access points for every game information.
2. the right part of the window displays the game properties and the tools used to modify them. Each panel in this part is linked with a node of the left panel tree. A property panel is generated by external Python modules and dynamically loaded. Each panel is based on a common template to be easily adapted to another type of game.

The tool bar gives the common tools to manipulate the game files (new, load, save...) and a language selector to add multiple languages to the games: the *blind-station* provides the possibility to display the game in multiple languages, the game generator must have the ability to edit the languages (see section 4.6).

4.4. *The property panel*

The property panel (see figure 4) is the base of the working interface, every information and tools are displayed on it. Its front-end and its behavior depend of the Python script used to generate it.

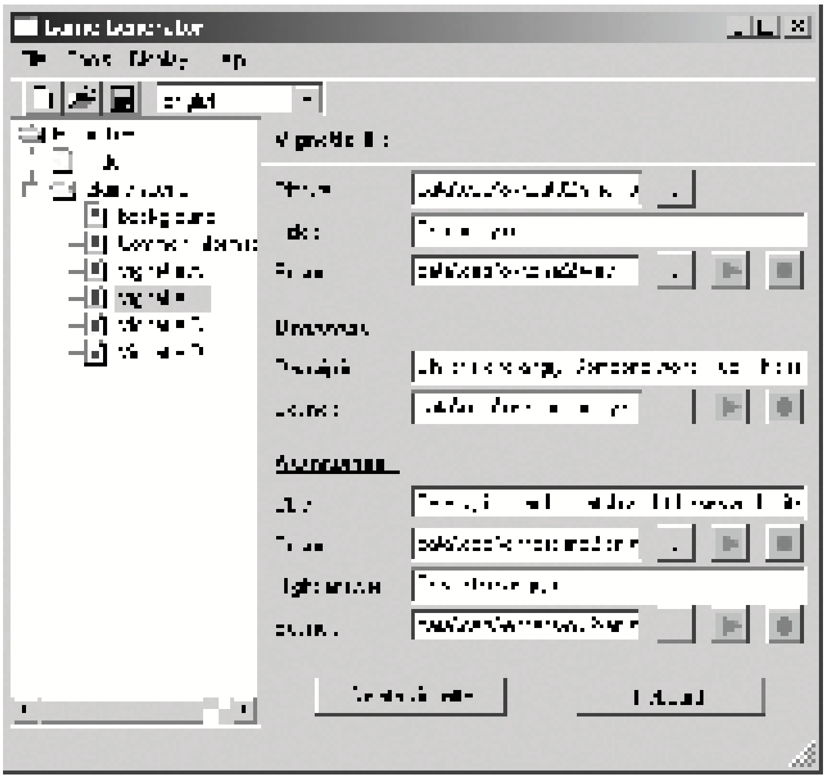


Fig. 4. The graphical interface of the game generator

Each panel is based on the same template but two types exist :

- the panels which provide information of one element (the item texts and sounds for example),
- the panels which broadcast their data to all the game elements (the item colors...). The difference is only data destination: for the first group of panel, the information are written in one XML location and for the second group, the information are stored in all the corresponding XML location. No difference is visible by the user (apart from the graphical result).

4.4.1. The item panel

Now let us have a look at the different panels. In the item panel, each graphical control allows modifying one property of the game:

- the selection of an image file changes only the image property (and duplicates the file in the game directory). The text control displays the picture real path, the final path (corresponding to the game directory) will be written automatically into the XML game file.
- when the user writes some text into a text control, the text is written directly to the XML game style sheet in the right position. The user writes the text in the language corresponding to the language selected in the tool bar.
- the game is based on sounds: the sound messages can be recorded and played directly with the corresponding tools (the red circle to record and the green triangle to play, see figure 4). If the sound file exists, the user can select it (like a picture and with a copy of it into the game directory).

4.4.2. The virtual tactile board

findit! is a tactile game, the user must define the right size and position of each tactile shape on the tactile board. A specific tool, accessible via the button “keyboard” of the item panel, is dedicated to this task. Figure 5 presents the tactile board configuration dialog. The board area is divided into 16 X 16 sensitive keys symbolised by the rounded rectangles on the dialog box:

- light grey cells are available (the currently unused tactile area),
- dark grey cells are already used by another item than the current one,
- black cells are the ones selected for the current item tactile area.

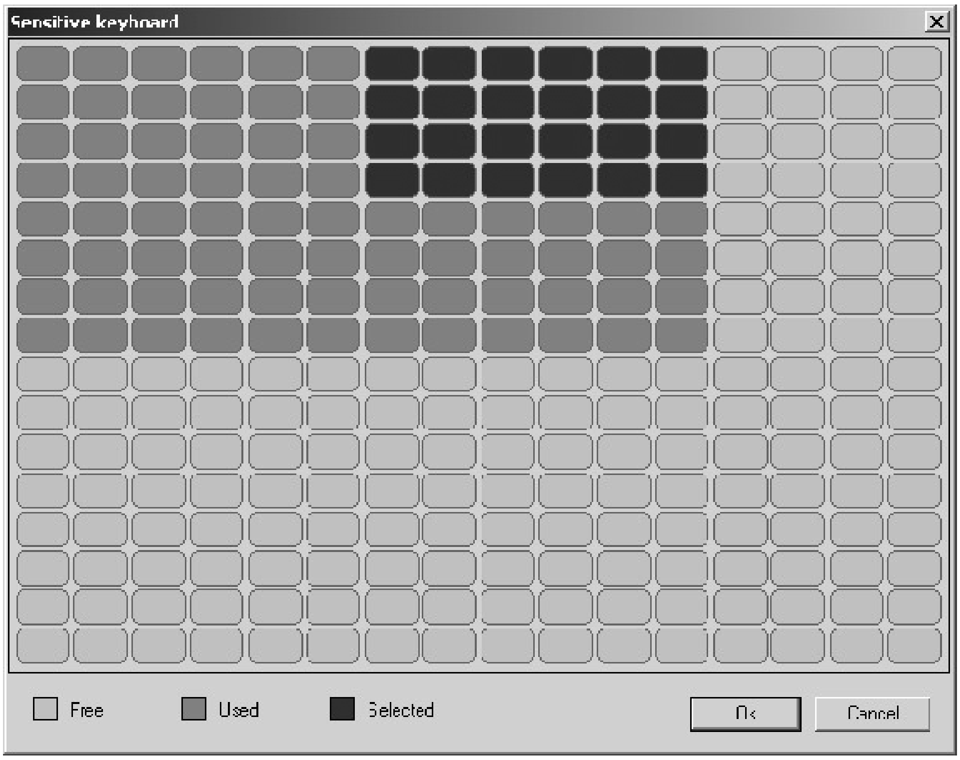


Fig. 5. The generator virtual tactile board

To change the area allocated to a item, the user can select all the rounded rectangle one by one or draw the area by a "click and move" action with the mouse. Multiple selections and complex shapes are allowed for the tactile area but the areas corresponding to two different items of the game cannot overlap.

4.4.3. The color panel

In the color panel (see figure 6), multiple properties can be changed together: the color selector modifies multiple color properties at the same time to maintain a game style unity. In this panel, if the user changes the color corresponding to the element described by the left label, the generator changes automatically each information according to the rules given by the XML configuration file to the XML game file and to the game display.

4.4.4. The game design

The position and size of each element (text, picture, menu) in the game can be modified graphically with the mouse, no control is placed on the property panels for them. To simplify the graphical interface, all these controls are placed in another window: the game preview.

4.5. The game preview

When a new template is created, the user wishes to be able to see the actual result directly without having to save the game and load it into the *blindstation* game engine. To do that, the generator displays a game preview: this is a separated window almost identical to the real game view (see figure 7).

By default, this window is displayed, and it is accessible from the "Tools" menu (if it is closed for any reason). The preview is near the game generator display. For the time being, only the text font and its size are different. Each modification from the property panel is automatically reflected to the preview window.

The preview is not a static window, this is a powerful tool to manipulate the game design. Every element can be resized an moved with a mouse drag and drop or with the numeric controls (to write numerically the right position and size): the menu options and the subtitle bar, the item legend and its picture. To help the user, some alignment tools are available to work with a group of objects.

The object selection is done by a mouse left button click on it, multiple selection uses the same process: the user can select as many elements as he/she wants. To

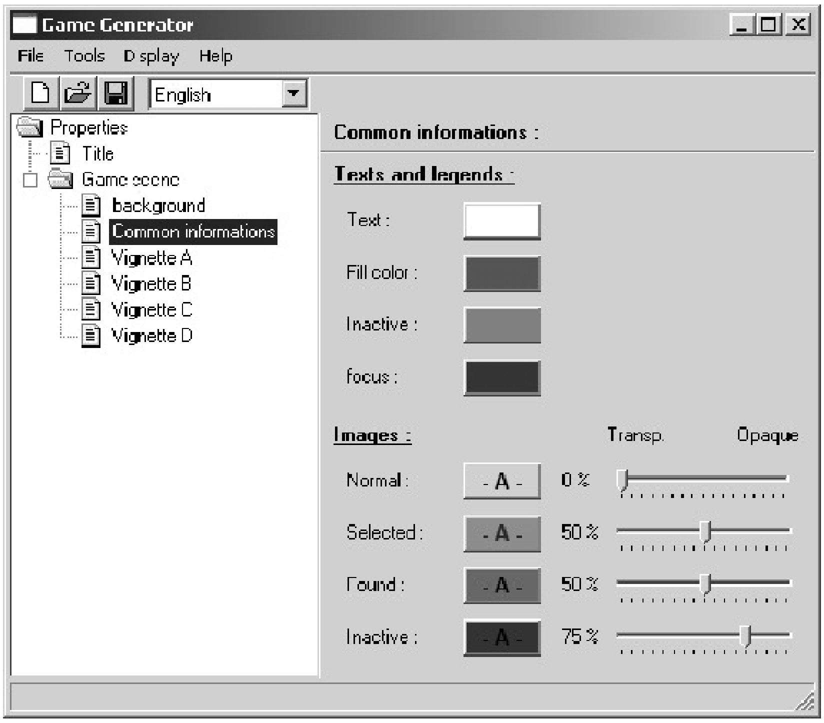


Fig. 6. The color panel of the generator

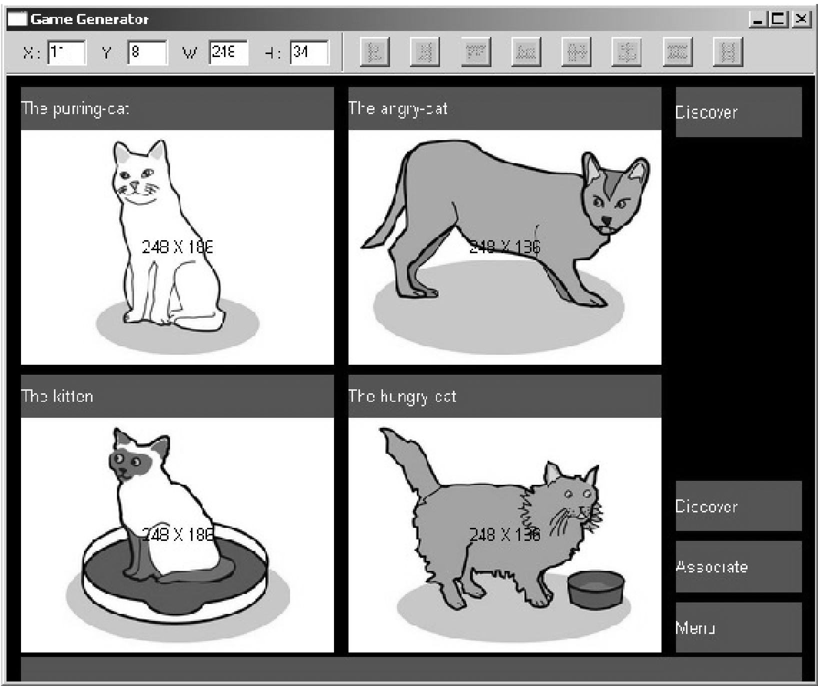


Fig. 7. The game preview windows.

unselect the object or the group, he/she just clicks the right mouse button anywhere in the preview window.

Using the preview window tools depends on the element selection:

– one selected object: the user can change the size and position with the mouse (click and drag the corresponding black square on the object control). He/she can write this information in the tool bar

text controls to increase the design accuracy.
- two or more selected objects: he/she can align the group of objects (to the left, right, top or bottom) according the first selected object. He/she can resize or center the group horizontally or vertically.

Text and color cannot be edited in this window, to do that, the user must use the property panel in the main window. But all the texts are displayed in the correct language according to the selection in the main window.

4.6. The generator and game languages

The textual information are not the most important part of a tactile/audio game but a correct treatment of the text in the various languages (the *blindstation* is multilingual) makes the game user-friendly (for a sighted person). This is very different for the generator, the user is necessarily a sighted person who has a preferred language. The software must display the most suitable language according to the user selection.

According to the last remark, two types of language are used in the generator, the first is the generator language for the graphical interface, and the second is the game language.

4.6.1. The generator language

The generator languages are the most important to the generator user. They are very easy to manage because the generator interface is static for one type of game (the associative game like *findit!*). The texts are translated and stored in a specific XML file named the “translation file” (see figure 8). This file contains all the sentences needed by the generator front-end and by the Python modules used to generate the property panels.

If a new language must be added into the generator, the translation file must be edited with an external editor. No specific tool is needed to do this work but no edition capacity is provided by the generator to do it.

The language of the generator graphical interface can be selected via the “Preferences” item of the “Tools” menu. The user can do this selection once for all, the software stores it.

4.6.2. The game languages

The game languages are different: the generator was created to assist the user to write this sentences. The game language editor is a part of the generator tools. A language which must be edited is selected from the drop-down control in the generator tool-bar. After that, the user writes his sentences in the property panel.

```
<menu_file_open>
 <en>&amp;Open...&#x09;Ctrl-O</en>
 <fr>&amp;Ouvrir...&#x09;Ctrl-O</fr>
</menu_file_open>
<menu_file_open_comment>
 <en>Open an existing game</en>
 <fr>Ouvrir un jeu existant</fr>
</menu_file_open_comment>
<menu_tools_preferences>
 <en>&amp;Preferences...</en>
 <fr>&amp;Prfrences...</fr>
</menu_tools_preferences>
<menu_tools_preferences_comment>
 <en>Modify software parameters</en>
 <fr>Modifier des paramtres du logiciel</fr>
</menu_tools_preferences_comment>
<propvignette_msgonevignette>
 <en>It isn&apos;t possible to delete the last vignette !</en>
 <fr>Il n&apos;est pas possible de supprimer la dernire vignette du jeu !</fr>
</propvignette_msgonevignette>
<propx_msgimage>
 <en>Select an image file</en>
 <fr>Choisir un fichier image</fr>
</propx_msgimage>
```

Fig. 8. A part of the XML translation file

Whatever the language, the user will always write the same sentence at the same place. The game texts are stored into the XML game style sheet directly with all the translations. To find the correct language from the XML game file, the *blinstation* uses the XML “lang” attribute.

4.7. The advanced sound recorder

The association game *findit!* is based on the sound but the sound controls (play and record) are very light in the generator property panel. A more complete interface is included inside the generator to provide new functions.

Figure 9 presents the advanced sound recorder, several tools are available:

1. The sound files can be loaded from and saved to any location with the “.wav” sound format. This is not a restriction because the *blindstation* framework and more precisely the associative game *findit!* can play only this format.
2. The recording can be started and paused at will to help the user to manage his audio subject.
3. Some sound parameters are editable:
   - the recording and playing sound level,
   - the sampling rate of the sound recording,
   - the input and output peripherals, if more than one of each exist.

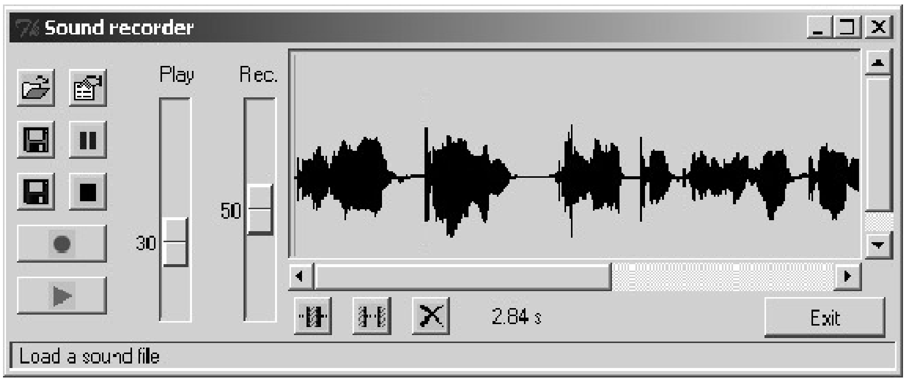


Fig. 9. The advanced sound recorder interface

4. A sound wave display can be used to select and transform a sound block (the block is defined by the left and right mouse click to fix the start and end of the block, a double click deletes the block marks). This block can be played alone and two cutting tools are available:
   – the first remove the sound bloc,
   – the second remove all but the sound block. This tool is useful to cut the start and end blank sound after a recording.

This tool completes the generator by adding a graphical display, some cutting tools and other functions. On the current version, the advanced sound recorder is accessible only from the menu. But it would probably be more efficient to have a direct link to it in the property panel. The technical incompatibility between WxWidgets (generator GUI) and Tk (Snack GUI) prevents a direct integration of the advanced recorder to the generator.

5. Conclusion

The generator is a versatile tool to create tactile/audio associative game content, it is easy to adapt it to build new types of games. To do that, the game creator must change the XML conversion file and create new external modules to display the tools and edit the properties. A Python script knowledge is required to extend or create a new type of game and the needed files for the generator (XML conversion file and front-end modules) but the process is simplified by the module template, a very common base to develop or adapt modules.

To finalize this tool, we must fix the text font display problem to create a more realistic preview of the game and translate the generator into more languages (only French and English full translation exist today, a swedish version is currently under development).

An extension of the generator is currently being developed which allows to link templates, so one is played when the previous one is done. This extension allows also to determinate in what mode each template must be played (Discovery/Association)

In order to facilitate sharing of overlays and templates, we intend to develop a web repository for templates and interface to download easily a new template from this web server.

6. Acknowledgments

The *TiM* project have been funded by the European Commission[7], on the program IST 2000 (FP5/IST/-Systems and Services for the Citizen/Persons with special needs), under the reference IST-2000-25298.

The TiM project participants are : *Inserm U483 INOVA, Université Pierre et Marie Curie* (France), coordinator, *Les Doigts Qui Rêvent* (France), *Université du Havre* (France), *Association BrailleNet* (France), *Royal Institute of Technology* (Sweden), *Sunderland University* (United Kingdom), *Swedish Institute for Special Needs Education* (Sweden) and *Virtua Ltd* (United Kingdom).

7. References

[1] D. Archambault. The TiM Project: Overview of results. In K. Miesenberger, J. Klaus, W. Zagler,

[7]The contents of this paper is the sole responsibility of the authors and in no way represents the views of the European Commission or its services.

and D. Burger, editors, *ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs)*, pages 248–256. Springer, 2004.

[2] D. Archambault and D. Burger. From Multimodality to Multimodalities: the need for independent models. In C. In Stephanidis, editor, *UAHCI'01 (Universal Access in Human-Computer Interaction, joint with 9th International Conference on Human-Computer Interaction)*, pages 227–231. Lawrence Erlbaum Associates, 2001.

[3] D. Grammenos, A. Savidis, and C. Stephanidis. UA-chess: A universally accessible board game. In *HCI: Exploring New Interaction Environments*, Las Vegas, Nevada, USA, 2005.

[4] A. Mitchell and C. Savill-Smith. The use of computer and video games for learning. The Learning and Skills Developpment Agency, London, 2001.

[5] R. Raisamo. Haptics research in tauchi: Aiding visually impaired children and enabling haptic interaction. In *Hands on Haptics: Exploring Non-Visual Visualisation Using the Sense of Touch - CHI 2005 workshop*, Portland, Oregon, 2005.

[6] S. Sablé and D. Archambault. Blindstation: a Game platform adapted to visually disabled children. In G. Craddock, L. McCormack, R.Reilly, and H. Knops, editors, *Assistive Technology – Shaping the future, Proceedings of the AAATE'03 Conference*, pages 232–236. IOS Press, 2003.

[7] Calle Sjöström. The sense of touch provides new computer interaction techniques for disabled people. *Technology and Disability (IOS Press)*, 10(1), 1999.

[8] T. Westin. Game accessibility case study: Terraformers a real-time 3D graphic game. In *ICDVRAT'2004*, pages 95–100, Oxford, UK, 2004.

Chapitre 5

Accessibilité des jeux vidéo

Finalement ce dernier chapitre comprend tout d'abord deux articles dans lesquels nous tirons quelques règles d'accessibilité des jeux vidéo. Ces règles ont été tirées de l'expérience accumulée lors du projet TiM et des nombreuses sessions d'évaluation qui y ont pris place. Enfin dans le dernier article nous décrivons un modèle, inspiré du modèle permettant l'accessibilité des applications dites de bureau (suite bureautique, mail, navigateur Web), qui nous permettra d'avancer plus rapidement vers la mise en accessibilité d'une majorité de jeux grand public. Ce modèle basé sur d'une part l'analyse de besoins en données des interfaces spécifiques. Il est basé sur une interface logicielle permettant au jeu d'exposer des informations d'accessibilité à des interfaces de jeu spécifiques.

Nous avons soumis en 2008, en réponse à un appel d'offre de la Commission Européenne, une proposition de projet sur ce sujet (de type IP – *Integrated Project*), qui n'a malheureusement pas été retenue. Nous sommes sur le point de proposer une version retravaillée à l'occasion du nouvel appel.

5.1 [HCI'2005b]

Titre	**Computer Games that Work for Visually Impaired Children**
Auteur(s)	Dominique Archambault, Damien Olivier[a] et Harry Svensson[b]
Affiliations	[a] Université du Havre [b] The Swedish Institute for Special Needs Education, Resource Center Vision Stockholm – Suède
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of HCI International 2005 Conference (11th International Conference on Human-Computer Interaction)
Editeur(s)	Constantine Stephanidis
Lieu/Date	Las Vegas, Nevada, États-Unis, Juillet 2005
Pages	8 pages (proceedings on CD-Rom)

Voir mémoire sections **3.1.3**, page 29
et **3.4.2**, page 42

Computer Games that Work for Visually Impaired Children

Dominique Archambault[*], Damien Olivier[**], Harry Svensson[***]

[*] Université Pierre et Marie Curie
Inserm U483/inova – 9, quai Saint Bernard – 75252 Paris cedex 5 – France
Dominique.Archambault@snv.jussieu.fr

Abstract

This paper focuses on games that work for visually impaired children. Within the TiM project, several games specially adapted for blind children (from 3 to 10 years old) were designed, involving blind children to evaluate and improve the games. From these examples we try to extract a certain number of rules that help to make games that work for these children. This is more than games that are strictly accessible, that is games in which all contents can be accessed (directly or via an alternative), but games that are still considered as games by the targeted users: blind or severely visually impaired children from 3 to 10 years old!

1 Introduction

The mainstream commercial market for computer games and other multimedia products is extremely large and sighted children have a considerable experience of such games. Children who cannot use the ordinary graphical interface, because they are totally blind or because they have a severe visual impairment (sight rated <0.05), do not have access to this important part of the youth culture. Designing games that work for visually impaired children is quite a challenge since the main feedback channel in games is usually visual. Indeed even if audio is more and more used in mainstream games, it is only a complement in a huge majority of cases. For instance most of these games cannot be efficiently played without any sound!

1.1 Accessibility of games

Accessibility of games is a more complex problem then software accessibility in general. First, and it seems obvious, it is very important that accessible games still be games. Adults in work situation accept relatively big constraint on usability to be able to use the same software as their sighted work mates and to work on the same documents. For instance in a word processing software, the simple request to know if a word is in bold face or not (which is obvious for a sighted person), necessitates to select the word, and then to open a dialog box to check the properties of the characters. There exist other software solutions for word processing, like LaTeX, which allow much simpler access to this kind of information for a blind person, but they are usable only if the work mates of this person use it! Otherwise, it's globally more important to be able to use the same software than the others even if the usability is not so good. This is not the case with children, especially playing. In other terms it is not enough to find a technical way allowing to access to all information needed in the interface, the result must be as interesting and as usable as the original game.

Accessible games must still be games! (rule 1)

Additionally the visual display allows the user to have quick access to a large amount of information which can be dynamical. Usually sighted users select the relevant information at a glance. In the case of games based on fast reaction from the player, trying to give a description of the content of the screen (which is basically the function of screen reader software) would not be efficient enough to allow the player to react fast enough.

[**] Laboratoire d'Informatique du Havre EA 3219, Université du Havre - France
[***] The Swedish Institute for Special Needs Education, Resource Center Vision Stockholm - Sweden

1.2 The TiM project

Within the framework of the TiM Project (Archambault, 2004) we have studied various mainstream games dedicated to young children (3 to 10 years old), and we have carried out a study of the users needs. According to these results we have adapted some of them to be usable by children with severe visual impairment and we also have developed specific games for this group of children. These studies allow us to understand what is necessary for a game to be usable by children with severe visual impairment.

The TiM game engine (Sablé & Archambault, 2003) was developed to design games that work with any kind of device, specific devices (like tactile boards or braille devices) as well as standard devices (like keyboard, joysticks or gamepads). It has a multi-layer architecture, which main layers are: the platform API and low level modules. The low level modules are external libraries used by the platform internals to manage each kind of devices. The main goal of the platform API is to offer to game designers pre-existing functionalities to simplify game design. With the TiM game engine, the scenarios of the games are independent of the modalities used by the player: the scenari are implemented using the Python language and abstract objects (the objects of the API: buttons, sounds, etc). The data necessary to render each object of the scenario through the different modalities are described in a XML style sheet. A set of 7 very different games were developed that can be used as well by visually impaired and by sighted players. The only limitation is about hearing capacity, since those games are based on audio.

This paper will describe several very important issues that have to be addressed in a design for all process in the case of the design of games that would be intended to work for visually impaired young children. We will first focus on alternative modalities usable with visually impaired children: audio, tactile, enlarged screen views. Then we will explore several items concerning equivalence of modalities. These issues will be illustrated with examples from two games which have been developed within the framework of the TiM Project.

1.3 Cases studies

1.3.1 Reader Rabbit: Toddler

The first case study concerns the game "*Reader rabbit's Toddler*" (Buaud et al., 2002). We have developed an adaptation of this game intended to very young children (approximately from 3 to 5 years old). In the original version of this game, the player can have access to 9 different educational activities and a navigation menu allowing to choose and to launch the activities. Additionally there is an introduction animation and a closing one. In the activities the child can play with music, recognise various animals, listen to various songs, play with colours and shapes, etc.

From the 9 different activities available on the original version, 4 could be adapted. Obviously the introduction, closing and navigation menu were adapted too. Indeed an adaptation should keep the philosophy or the goal of the original game. It is sometimes possible to use some existing resources to make a game slightly different than the original one, because the goal of the original game is basically visual, but in that case it cannot be called and adaptation of this game. Indeed adapting any kind of content to another modality means to give access to the same content. Then, at it is often the case, it was not possible to make a relevant adaptation to all parts of this game.

The components of the games that were adapted are the following:

- *Navigation menu:* In a "magic environment" (figure 1), *Mat the mouse* guides the child. Around it the nine educational activities are represented by animated pictures. The cursor of the mouse is represented by a big star (just at the left of Mat on the figure). On the right bottom corner, a small star allows to quit the game.
- *Bubble castle:* The child should release the animals that are locked into bubbles. When the child released all the animals, the dragon (imaginary character of the game) counts the number of released animals.
- *Musical meadow:* Four sets of flowers represent musical instruments. By moving the cursor on each set the child makes the corresponding instruments play or mute. A second melody is played with 4 different sets of animals.
- *Baby basket bingo:* The child has to find the parents of the small animal hiding in the basket. Each baby plays a sound close to its parent's sound. The child has to move the basket toward the close to parent with the mouse.

- *Follow me theatre:* a rabbit sing 8 different songs. The child chooses a song by positioning the cursor of the mouse on one of the 8 pictures representing the songs that are displayed around the rabbit.

The adaptation of this game was designed to be usable without any visual information. Then all output was converted to alternative modalities (basically audio in this game) and all input must be accessible via the keyboard or a tactile interface.

The tactile interface is based on a specific device called tactile board. It's an input device on which a rectangular sensitive area is divided into 256 cells. On top of the sensitive area, a tactile overlay may be inserted. A rich tactile overlay was designed, using pieces of various materials, stuck on a robust PVC sheet, and Braille labels (figure 2).

Figure 1: *Reader Rabbit Toddler*
(the "magic environment" in the original game)

Figure 2: *Reader Rabbit Toddler*
The tactile overlay

This adaptation was developed according to a user centred design process, involving children at different steps. Several prototype versions (of the software and of the tactile overlay) were evaluated with them, before we could release the actual version[1].

1.3.2 Mudsplat

Mudsplat is an arcade like game where the player defeats mud throwing monsters by squirting water at them. This game was designed according to the basic idea of the classic arcade game "Space Invaders": a lot of enemies appear in front of the player. With a weapon, the player has to fire and defeat those enemies. Designing a game that can be played without sight lead us to a completely audio game. The player can hear the monsters appearing on a line in front of him; he/she estimates their position (thanks to stereo pan) and has the possibility to move left or right (with keyboard or gamepad); then when he/she thinks a monster is in front of him, he can shoot with a water hose: the monster disappears (it is well known that monsters don't like to be clean!). The monsters can also fire throwing mud (if hit, the player will loose one life).

Arcade games were originally commercial computer games that were played in arcades. The increasing difficulty of gameplay allow beginners to have fun on easy levels, while skilled player can play for a long time before losing. The actual gameplay is often quite repetitive (and then easy to learn), but the player is motivated by a continuously increasing difficulty, by achieving a high score and by collecting different bonus objects. Arcade games are divided in different worlds and levels and the overall goal is to finish the very last level. There is often a story that explains the different worlds and levels.

In Mudsplat the player can play 25 levels starting very easy, where the number of simultaneous monsters increases, as well as new kind of monsters more malicious. The main traditional features of arcade games were implemented (like extra lives, levels, bonus objects, bonus levels,... A high score list is maintained where the player may insert his name.

[1]At the time of writing, the French version is ready to be distributed as soon as we complete successfully negotiations with the owner of the original game. Information will be available on: "http://www.timgames.org".

By maintaining a very low level of necessary keys used for input we aim for this game to be playable for younger children. With proper introduction, 5-6 year old children can start to play and understand the interface and gameplay. The gameplay starts on a very easy level, and the degree of difficulty increases during the game, so a younger child will probably succeed at once, but will eventually need to practice to become good at playing the game. There is also a full help menu included in the game with explanations to all the sounds used in the game. Furthermore there are no gender references in the story or the speaker texts and our approach has been to make a game that could attract both boys and girls.

2 Alternative modalities usable with visually impaired children

2.1 Audio representations

The relevance of sound representations is a very big concern. In many mainstream games, we have observed that a lot of sounds are understood thanks to the visual context. For instance if a lion is on the screen, even a very bad roaring sound record will be eared as the roaring of a lion, while the same audio file played without visual help cannot be interpreted. In that case there are two possibilities.

First we think it's preferable as far as possible to use sounds that are really relevant by themselves, like real recordings. In the adaptation of the game “Reader Rabbit Toddler”, we have replaced the original sounds of animals by recordings of real animals. But this is not always enough because it is not obvious that a blind child have ever heard a real lion, and is able to make a difference between the roaring of a lion and of a tiger. In our case it was relevant because there were no ambiguity between the different animals present at the same time.

This leads to the second way of handling that problem, which can be combined with the first one. Using other modalities, we have to provide some additional contextual information to replace the visual context which is missing. This information should be designed in order to give to the child enough clues to recognise the different sounds.

Ensure that the sounds can be efficiently recognised by children,
by using relevant sounds or providing contextual information. (rule 2)

2.2 Tactile representations

There are two ways of using the tactile modality. The first one is to use a Braille display. This is essentially made to display text and is not suitable for very young children. However we are currently conducting researches to use Braille display in action games. The other possibility, which will be discussed here is to use tactile overlays on a tactile board. This allows to use very rich tactile representations.

Indeed tactile overlays may be prepared using various technologies. In order to reach the best quality level, the tactile overlay for the game “Reader Rabbit Toddler” was designed on a PVC sheet, very robust, with stuck pieces of various material (like rubber, fabrics, plastic, leather, fur, etc...) and Braille labels laid down with a new technology called TOM3D[2]. It is also possible to use thermoformed overlays, swallowed paper (Zytech™), or simply ordinary paper with stuck elements.

The overlays are inserted in a tactile board like the one described in section 1.3.1 about the game “Reader Rabbit Toddler”. This device needs to be driven directly by the software (the TiM game engine supports it). Other kind of alternative tactile boards, like Flexiboard™ or IntelliKeys™ may be used. These tactile boards are connected on keyboard port so it is important that all functions of the game can be accessible using keyboard shortcuts. Additionally this will ensure that the game is accessible via various devices or software systems like switches or keyboard scanning software.

All functions of the game must be accessible using keyboard shortcuts. (Rule 3)

[2]TOM3D was developed by LDQR, the unique publisher of tactile books in Europe: http://www.ldqr.org

The major problem with this kind of interface is that is that the overlays are static. On the counterpart they can be designed in a very rich way and be very attractive to children. Then we essentially use this interface for games intended to the younger children (basically from 3 to 6/7 years old). Additionally, whatever model of tactile board is used, changing the overlay is not easy. It may be because of the fact that insertion of the overlays is not easy or because it's not possible to bind together the different overlays (and then the problem comes from the selection of the overlay). Then we prefer to limit the number of overlays for the same game. If possible, the whole game should be played using the same overlay. The only exception would be about games where the goal itself is to discover the overlays.

Limit the number of overlays for the same game. (rule 4)

For the same reason as above (this interface is intended to very young children), it is important to limit the number of buttons on the overlay, in order to lessen the memory load of the child. Obviously the functions of the buttons should be easy to remember.

The actual overlay for the game “Reader Rabbit Toddler” contains 10 buttons. These buttons could be described in two parts:

– navigation buttons : “GAME” (to scan the different possible choices), “ENTER” (to select one of theses), “HELP”, “EXIT”, “YES” and “NO”. These buttons corresponds to common computer interface tasks, so we decided to keep the common computer terminology as far as possible.
– play buttons used in the different activities (they were 4 of these, that we called “PIANO” buttons).

Limit the number of buttons on the overlay. (rule 5)

When the game is started, it cannot be assumed that the player is familiar with the tactile overlay. Then it's necessary to describe it. In order to make it interesting to the child, and also to be really efficient, this description should be interactive. But as the overlay is static, we cannot have a character showing the different elements like on a screen. In the game “Reader Rabbit Toddler”, Mat the mouse will describe the tactile feeling of each button and ask the player to find it and gives his name and a description of its function in the game. In the different activities some additional explanations may be given, reffering to the given name of the button.

Exemple: *“Look for the big button, very soft with long hair. It is the GAME button. When you press it, you will hear the names of the different games.”*

On the other hand it would be boring for more experienced users to have to go through this discovery at each time. Actually Mat the mouse proposes to start directly to play by pressing the “Enter” key. If it is not pressed soon, then the discovery mode is started.

Design an interactive discovery mode for the tactile overlays. (rule 6)

2.3 Enlarged graphical view

For children who have a profound low vision it is necessary to have interesting things on the screen. The TiM game engine allows to display in the same time a graphical view (see section 3.4) and enlarged texts or pictures. The enlarged screen is considered as another display, like if 2 different screens were connected, with a “zoom” key allowing to switch from one to another. Then it's possible to stay in zoom mode or to switch from one view to the other all the time, depending on the player's needs and the people who may play with him. The zoom mode can be tuned according to the sight of the player (sizes, colours).

Provide enlarged graphical views when possible, that can be tuned according to one's sight. (rule 7)

In the game “Reader Rabbit Toddler”, the introduction and the closing are cartoon animations. This must be avoided with most partially sighted children who cannot see moving images. Instead, we extracted several pictures of the animations to propose a kind of slide show, synchronised on the audio. In games where it would be relevant (for instance if the understanding of images have an importance in the game), this could be even improved by letting the child the possibility to pause and resume it (but then an audio description should be available to for the blind).

Give alternatives to video and animations. In most cases the audio only is not enough. (rule 8)

In scenes in which navigation is needed, it might be interesting to give to some partially sighted player an help to locate where he/she is in the screen. For instance in some of the activities of the game “Reader Rabbit Toddler”, like “Baby basket bingo” (see figure 3), the currently focussed element is framed using a 50% opacity red rectangle.

Figure 3: Baby basket bingo

3 Equivalence of modalities

3.1 Usability and difficulty level

When designing a game that can be used with several modalities, a very important issue is to ensure that the usability is the same irrespective of the modality used by the player. The player must be able to choose the modality he will use according to his possibilities and not according to his level. In the game “Reader Rabbit Toddler” it is possible to use as well a tactile board as an input and the standard keyboard. Each command of the tactile board corresponds to a key of the keyboard. This allows to use the games with alternative keyboards, switches etc.

When the score is an important issue in the game, it is necessary to check if the difficulty level is the same for all players, so the scores can be comparable, or alternatively when it is not possible, to append to the score the way it was obtained. In the game “Mudsplat” it is possible to use the keyboard as well as a joystick (or gamepad) to play, and the difficulty is comparable. On the other hand, the output is only audio. Indeed the tests we did with visual information about the position of the monsters showed immediately that it was enormously more easy to play with the sight. Indeed the sight would allow to see in a glance tenths of enemies or more (like in the original Space Invaders game), while the hearing only allow to discriminate a few of them (in Mudsplat a maximum of 5 different monsters and bonuses objects can be heard simultaneously, together with background sounds and music).

Ensure that players using one modality would not be favoured. (rule 9)

3.2 Relevance of the guidelines in a multimodal context

In a multimodal context, it is important that the guidelines allow the player to understand easily how to do with any of the available modalities. This can be realised by having different guidelines for different modalities, but when it is possible it is preferable to refer to names (of buttons for instance) that have a cross modal meaning (in the game) so the same explanation fits all.

Depending on the age of the target group, this is not always possible. For instance the simple sentence “Click on something” cannot be used on a tactile board if the game is intended to very young children, and it is better to say “Press on something” then.

Check the relevance of the guidelines for each modality.(rule 10)

3.3 Interaction and game play

It is very important that players allways have an immediate feedback after each action. For instance in the game Mudsplat, the usability was improved when sounds were added to the move of the player. Actually when the player moves, all his environment is shifted to the left or the right (according to the direction of his move) but it is not immediate. Giving an immediate sound feedback (it's a very soft foot step sound) improved very efficiently the game play.

In the menus of the same game, the labels of the the items are said (by a recorded voice). It takes a few seconds on each. A very short "toc" sound was added just before so the player can move faster if he/she already knows the game.

Immediate non visual feedback (usually audio) after any input.(rule 11)

When the game starts, it cannot be assumed that the player is already familiar to it. A first solution could be to record on the disk information about how many times the game was started since installation but still it could be different players on the same machine. The player cannot neither see on the screen where to click to have some help. It is necessary then to have some help coming automatically by default, that can be stopped by experienced users. It was already explained about tactile interfaces (see section 2.2, rule x).

In the case of "Mudsplat menus", when the menu is started, the player hears "Main menu", followed by a short pause if he/she did not start to move into the menu, he/sje will hear "Use arrow keys to navigate and press enter to confirm". This is immediately stopped too if the player starts to move during it.

Give audio help by default, with a way of stopping it by experienced users. (rule 12)

3.4 Graphical view

The TiM game engine allows to display a graphical view. It is very important for the for the relation with sighted people close to the blind player (brothers and sisters, school mates, parents). The player can then refer to what's on the screen. Depending on the game this information may allow to play (like in "Reader Rabbit") or not. In "Mudsplat", as explained in section 3.1, it would be too easy with the screen for sighted users, then still images are displayed on the screen during the game play (see figure 4).

Display interesting contents for the sighted on screen. (rule 13)

Figure 4: Mudsplat screen during game play

3.5 Compensation of visual information

In many games, the rewards (when the player succeeds) have audio and visual elements. We have observed that the audio may be too poor to make it interesting enough. In the game "Reader Rabbit Toddler", we designed a first version with a tactile board and allowing "technically" to complete several activities. The rewards were all the audio of the original game. This version was tested with several blind children and we could observe that they managed to learn how to use the game and to succeed in the different activities. But after a while, they would ask: "Is the test finished? Can I go to play now?". So they were not considering that the game was actually a game.

After that we asked an author (who usually write stories for children books or magazines) to write some kind of poems corresponding to the various cases. These poems were recorded and inserted at the start of the activities, and as rewards when relevant, etc. For instance in the Baby basket bingo, when the player succeeds to find the mother of the foal, he will hear:

« La maman est ravie	The mother is happy
Et le bébé aussi.	And so do the baby
La jument et son poulain	The mare and her foal
Vont se faire plein de câlins. »	will cuddle each other a lot

There is a text for each animal (with the 2 same first lines – can be mother or father thought).

The next version (with poems) was then evaluated and after a while most of the children did ask of they could have a version of the game at home!

Check if reward and pleasant things are enough fun without the visual. (rule 14)

4 Conclusion

Designing games that work for severely visually impaired children implies high constraints. Indeed it may be relatively easy to make accessible a piece of software in the meaning of ensuring that users can access to all the contents. In the case of games, two additional constraints are (1) that the access to the relevant information comes at the good time to ensure that the player can actually play, and (2) that it is still fun! In a design for all perspective, these issues are crucial for the visually impaired young children.

Acknowledgements

The TiM project was funded by the European Commission[3], on the program IST 2000 (FP5/IST/Systems and Services for the Citizen/Persons with special needs), under the references IST-2000-25298. The TiM project participants are: *Inserm U483 INOVA, Université Pierre et Marie Curie* (France), co-ordinator, *Les Doigts Qui Rêvent* (France), *Université du Havre* (France), *Association BrailleNet* (France), *Royal Institute of Technology* (Sweden), *Sunderland University* (United Kingdom), *Swedish Institute for Special Needs Education* (Sweden) and *Virtua Ltd* (United Kingdom).

We thank the company "*Mindscape*" for allowing the adaptation of the game "Reader rabbit's Toddler".

References

Archambault, D. (2004). The TiM Project: Overview of Results. In K. Miesenberger, J. Klaus, W. Zagler and D. Burger, (Eds.), *Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs)* (pp. 248-256). Berlin: Springer LNCS 3118.

Archambault, D. and Burger, D. (2000). TIM (Tactile Interactive Multimedia): Development and adaptation of computer games for young blind children. In *Proc. ERCIM WG UI4ALL & i3 Spring Days 2000 Joint workshop, Interactive Learning Environments for Children*. Athens.

Archambault, D. and Burger, D. (2001). From Multimodality to Multimodalities: the need for independent models. In C. Stephanidis, (Ed.), *Towards an Information Society for All, Proc. of the UAHCI'01 conference (Universal Access in Human-Computer Interaction).* (pp. 227-231). Mahwah, New Jersey: LEA.

Buaud, A., Archambault, D. and Roussel, B. (2003). Ergonomic evaluation of computer games for visually impaired children. In C. Stephanidis (Ed.), *Universal Access in HCI - Inclusive Design in the Information Society*, vol. 4 (pp. 1265-1269). Mahwah, New Jersey: LEA

Buaud, A., Svensson, H., Archambault, D. and Burger, D. (2002). Multimedia games for visually impaired children. In K. Miesenberger, J. Klaus, W. Zagler (Eds.), *Proc. of the ICCHP 2002 (8th International Conference on Computers Helping People with Special Needs)* (pp. 173-180). Berlin: Springer LNCS 2398.

Sablé, D. and Archambault, D. (2003). Blindstation: a Game platform adapted to visually impaired children. In G. Craddock, L. McCormack, R. Reilly and H. Knops (Eds.), *Assistive Technology - Shaping the future, Proceedings of the AAATE'03 Conference* (pp. 232-236). Amsterdam: IOS Press.

---

[3]The contents of this paper is the sole responsibility of the authors an in no way represents the views of the European Commission or its services.

5.2 [ACE'2005]

Titre	**How to make games for visually impaired children**
Auteur(s)	Dominique Archambault et Damien Olivier[(a)]
Affiliations	[(a)]Université du Havre
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of ACE05 Conference (ACM SIGCHI International Conference on Advances in Computer Entertainment Technology)
Editeur(s)	Newton Lee
Lieu/Date	Valence, Espagne, Juin 2005
Pages	450–453

Voir mémoire sections **3.1.3**, page 29
et **3.4.2**, page 42

How to Make Games for Visually Impaired Children

Dominique Archambault
Université Pierre et Marie Curie
Inserm U483/Inova
9, quai Saint Bernard
75252 Paris cedex 5 - France
dominique.archambault@upmc.fr

Damien Olivier
Université du Havre
Laboratoire d'Informatique du Havre
25, rue Philippe Lebon
76063 Le Have - France
Damien.Olivier@univ-lehavre.fr

ABSTRACT

This paper discusses the principal constraints encountered when adapting computer game so they work for visually impaired children. A game platform, the blindstation, was developed to answer to the technical problem. It allows to adapt existing content or create some new games. It provides a set of Python functions to describe those games in an abstract way, independent from their representation. The platform can then render the game in a multi-modal way using the screen, keyboard, mouse and joystick, but also using some specific devices like a Braille terminal, 3D sound, a tactile board or a speech synthesiser. The rendering is done according to an XML style sheet which describes the available resources. It can be customised depending on the available devices but also on the user's choices and disabilities. Our experience is based on the TiM project intends to develop and to adapt computer games for visually impaired children. Several games specially adapted for blind children (from 3 to 10 years old) were designed, involving blind children to evaluate and improve the games. We have established that this is more than games that are strictly accessible, that is games in which all contents can be accessed (directly or via an alternative), but games that are still considered as games by the targeted users.

Keywords

Visually Impaired, Multimedia, Multi Modality, Computer Games

1. INTRODUCTION

The mainstream commercial market for computer games and other multimedia products is extremely large, sighted children have a considerable experience of such games. Children who cannot use the ordinary graphical interface, because they are totally blind or because they have a severe visual impairment (sight rated < 0.05), do not have access to this important part of the youth culture [5, 4]. Designing games that work for visually impaired children is quite a challenge since the main feedback channel in games is usually visual. Indeed even if audio is more and more used in mainstream games, it has only a complementary role in a huge majority of cases. For instance most of these games can be played efficiently without any sound!

Accessibility of games is a more complex problem then software accessibility in general. It seems obvious, but it is very important that accessible games still be games. We need to develop accessible interfaces but also to take into account the consequences of the visual deficiency regarding to the usability (for instance the time needed to actually access to all the information provided in alternative modalities). Several very important issues have to be addressed, in a design for all process, to design games that would be intended to work for visually impaired young children. These aspects will be considered in this paper.

2. TIM PROJECT

The purpose of this project is to offer multimedia computer games intended for young children who are blind or severely visually impaired [3, 2]. TiM is also addressing visually impaired children with additional impairments in form of slight to moderate degree of learning or cognitive disability and/or a physical impairment. These games are planned to be used by the children in an autonomous way, without assistance of a sighted person. To reach the needs of these children, the games have to be adaptable to their specific needs and to all the specific devices they use, each corresponding to a specific modality: tactile boards, Braille displays, speech synthesisers, as well as standard devices like keyboards and sound devices or adjustable screen settings (scalable font prints, customisable colours and contrast...).

3. ALTERNATIVE MODALITIES

Visually impaired children who cannot use the standard graphical display can play computer games using alternative modalities, so called non visual modalities. They are based on auditory and tactile senses. In this section we discuss the modalities that have been used in the TiM project, focussing on the constraints they impose on the game development. The main property shared by all these non visual modalities is the absence of global perception: they are intrinsically sequential.

3.1 Audio representations

The relevance of audio representations is a very big concern. In many mainstream games, we have observed that a lot of sounds are understood thanks to the visual context. For instance if a lion is on the screen, even a very bad roaring sound record will be heard as the roaring of a lion, while the same audio file played without visual help cannot be interpreted. In that case there are two possibilities.

- First it's preferable as far as possible to use sounds that are really relevant by themselves, like real recordings.
- The second way of handling that problem can be combined with the first one. We have to provide some additional contextual information, via other modalities, to replace the visual context which is missing. This information should be designed in order to give to the child enough clues to recognise the different sounds.

3.2 Tactile modalities

There are two ways of using the tactile modality.

- The first one is to use a **Braille display**. This is essentially made to display text and is not suitable for very young children. In the TiM project, it can be used to display text of menus or commands, and help text. However we are currently conducting researches to use Braille display in action games.
- The other possibility is to use **tactile overlays** on a tactile board. This allows to use very rich tactile representations.

Tactile overlays may be prepared using various technologies. They are inserted in a tactile board like picture 1. This device is driven directly by the `Blindstation`. These tactile boards are connected on keyboard port so it is important that all functions of the game can be accessible using keyboard shortcuts. Additionally this will ensure that the game is accessible via various devices or software systems like switches or keyboard scanning software.

The major problem with this kind of interface is that is that the overlays are static. On the counterpart they can be designed in a very rich way and be very attractive to children. Then we prefer to limit the number of overlays for the same game. If possible, the whole game should be played using the same overlay.

3.3 Enlarged graphical view

For children who have a profound low vision it is necessary to have interesting things on the screen[1] . The TiM game engine allows to display in the same time a graphical view and enlarged texts or pictures. The enlarged screen is considered as another display, like if 2 different screens were connected, with a "zoom" key allowing to switch from one to another. Then it's possible to stay in zoom mode or to switch from one view to the other all the time, depending on the player's needs and the people who may play with him. The zoom mode can be tuned according to the sight of the player (sizes, colours). Animations must be avoided with most partially sighted children who cannot see moving images. Instead, we extracted several pictures of the animations to propose a kind of slide show, synchronised on the audio. In games where it would be relevant (for instance if the understanding of images have an importance in the game), this could be even improved by letting the child the possibility to pause and resume it (but then an audio description should be available to for the blind). In scenes in which navigation is needed, it might be interesting to give to some partially sighted player an help to locate where he is in the screen.

[1]We still consider this modality as non visual since it has the property to forbid global perception.

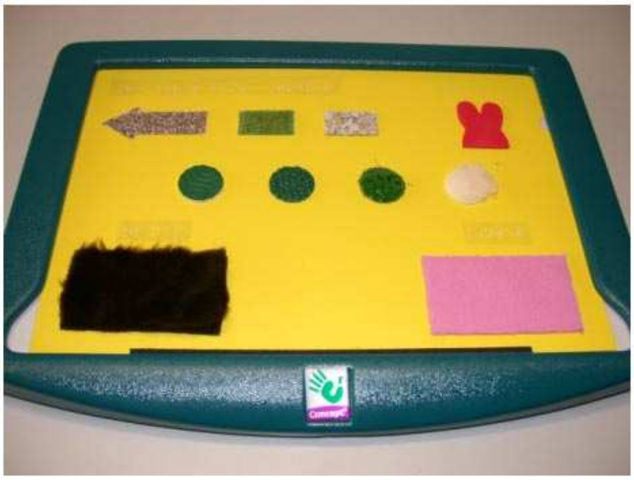


Figure 1: "Reader Rabbit Toddler", the tactile overlay

4. MODIFICATION OF THE INTERACTION PROCESSES

The game interaction processes must be modified in order to fit the modalities that can be used by visually impaired children, and especially focussing on their abilities.

- The modification often leads to the necessity to redesign important parts of the contents (see fig. 2). For instance when using tactile boards, it is not relevant to have help messages saying click on something. First because the action will be press and not click, and for very young player it might be important that the guideline is relevant. Another reason is that tactile board are used together with overlays on which are sticked different objects that are static, that is the objects will always be the same during all the game since on the screen they may change.
- It is very important that players always have an immediate feedback after each action. For instance giving an immediate sound feedback (it's a very soft foot step sound) improved very efficiently the game play.
- If adults in work situation accept relatively big constraint on usability to be able to use the same software as their sighted work mates and to work on the same documents. For instance in a word processing software, the simple request to know if a word is in bold face or not (which is obvious for a sighted person), necessitates to select the word, and then to open a dialog box to check the properties of the characters. This is not acceptable for a game, especially if it is intended for young children.

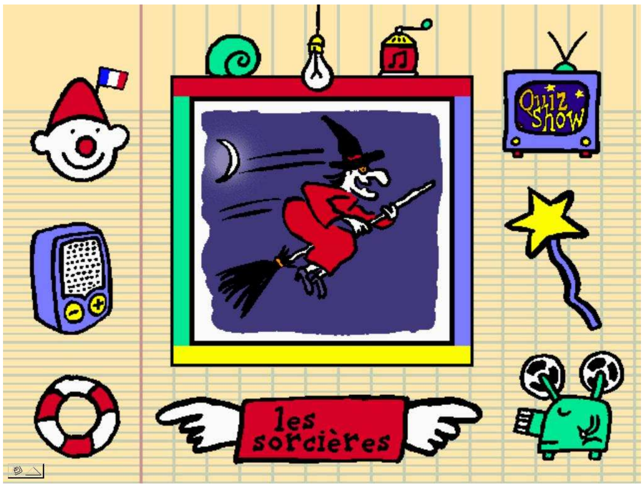
- As all the modalities we can use are sequential, it is very important that the player is provided the information needed at the very good time (for instance these modalities cannot let the player perceive the apparition of an enemy on the screen together with its position and features instantly).

Then making an adaptation of a game that work for this group of player necessitates a specific development. It is not enough to have a upper layer making th interface of the ordinary software and the mainstream game (like in the case of standard software like screen readers or Internet software).

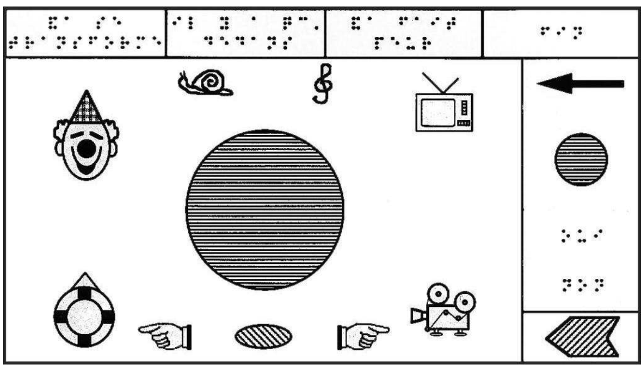
5. BLINDSTATION (TIM GAME ENGINE)

In order to simplify game creation and adaptation, a game engine or platform `Blindstation` [1], was developed.

The mainstream computer games are usually designed especially to be used through a standard multi modal interface (graphical display, mouse and speakers). So the approach for games development is based on visual conception: interaction objects of the game are generally represented in the design software as pictures which are attached to some parts of the screen.



(a) The original graphical interface



(b) The tactile adaptation

Figure 2: Adaptation

In order to be able to design games that are independent from the modalities and can use specific devices, the approach chosen by the TiM project is to use a modality-independent model. This is the reason why in the TiM platform, the game are described in 2 distinct parts:

- The game scenario is written in a script using some abstract high level components provided by the platform API. These components (buttons, sounds, etc) embed the way they will be rendered using the different modalities. The scenario is described in a game script written in the Python language (see for example Listing 1). The script is interpreted by the Python interpreter and uses functionalities of the platform through the API.
- the data necessary to render each component of the scenario (audio samples, music, texts, pictures, animations...) through the different modalities are described in a XML style sheet (see for example Listing 2).

This allows to separate the logic of the game and the data necessary to render it to the player.

The TiM game engine was developed to design games that work with any kind of device, specific devices (like tactile boards or braille devices) as well as standard devices (like keyboard, joysticks or gamepads). Low level modules (external libraries) are used by the platform internals to manage each kind of devices.

During game execution, the platform using the script and the style sheet will render the interface in a multi modal way according to the environment and the needs of the player.

Some components are used to describe the game flow and can be found in most traditional game platform. For example, the `Scene`, just like in movies, is a way to divide a game in many small coherent sequences; the `Game` component represents the context in which the whole game is running and is responsible for initialising many low level aspects of the platform (like loading the many libraries, initialising the display, the sound, the braille display...) and dealing with scenes changes.

6. CONCLUSION AND PERSPECTIVES

Within the framework of the TiM project, the Blindstation was used to design various different games [1], beyond which: Mudsplat (an arcade like game where the player has to shoot on everything he can ear. Using stereo, the objects are on a line in front of the player and he can move left and right); Reader rabbit's: Toddler (an adaptation of a mainstream discovery game designed for very young children); X-Tune (a musical construction game where the player can sequence and play with different sound environments); Tactile/Audio discovery (a very simple game for very young children or children with additional disabilities to to associate sounds and pictures on the screen or tactile information of a tactile board).

We have now the experience of adapting existing games, we can observe that the cost is higher than developing specific games, because most of the audio resources have actually to be redesigned. But it is very important anyway because it allows blind children to share their feelings and experiences of playing with their sighted peers. Another option is to design a specific game, reusing only a concept, ensuring that this game will be also interesting for the sighted, in a design for all perspective.

7. ACKNOWLEDGEMENTS

The TiM project was funded by the European Commission[2], on the program IST 2000 (FP5/IST/Systems and Services for the Citizen/Persons with special needs), under the references IST-2000-25298.

We thank the company *"Mindscape"* for allowing the adaptation of the game *"Reader rabbit's Toddler"*.
We thank the company *"Bayard Presse"* for allowing the adaptation of *"L'Univers de Pomme d'Api"*.

8. REFERENCES

[1] D. Archambault. The TiM Project: Overview of Results. In K. Miesenberger, J. Klaus, W. Zagler, and D. Burger, editors, *Proc. ICCHP 2004 (9th International Conference on Computers Helping People with Special Needs)*, Lectures Notes in Computer Science 3118, pages 248 – 256, Berlin, July 2004. Springer.

[2] D. Archambault and D. Burger. TIM (Tactile Interactive Multimedia): Development and adaptation of computer games for young blind children. In *Proc. ERCIM WG UI4ALL & i3 Spring Days 2000 Joint workshop, Interactive Learning Environments for Children*, Athens, Greece, Mar. 2000.

[3] D. Archambault, D. Burger, and S. Sablé. The TiM Project: Tactile Interactive Multimedia computer games for blind and visually impaired children. In Črt Marinček, C. Bühler, H. Knops, and R. Andrich, editors, *Assistive Technology – Added Value to the Quality of Life, Proceedings of the AAATE'01 Conference, Ljubljana, Slovenia*, pages 359–363, Amsterdam, The Netherlands, Sept. 2001. IOS Press.

[4] A. Buaud, H. Svensson, D. Archambault, and D. Burger. Multimedia games for visually impaired children. In K. Miesenberger, J. Klaus, and W. Zagler, editors, *Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs)*, volume 2398 of *LNCS*, pages 173 – 180, Linz, Austria, July 2002. Springer.

[5] A. Hildén and H. Svensson. Can All Young Disabled Children Play at the Computer. In K. Miesenberger, J. Klaus, and W. Zagler, editors, *Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs)*, volume 2398 of *LNCS*, Linz, Austria, July 2002. Springer.

[2]The contents of this paper is the sole responsibility of the authors an in no way represents the views of the European Commission or its services.

Listing 1: A simple example of a script using the Blindstation

```
from blindstation import *

class memory(Scene): # a simple memory game
 def __init__(self, game):
  Scene.__init__(self,game,'memo','init')

  self.cards=[]
  for card in ['card1', ...]:
   self.cards[card] = Widget(self, card, 'hidden')
   self.cards[card].validate_callback =
        self.validate_card
   ...
  current_card=None

  self.listener = Listener()
  self.source = Source("source")
  self.source.source_attach = self.source_callback
  # function called when a sound is finished

  self.source.queue("intro_music")
  self.source.queue("intro_speech")
  self.source.play()

 def validate_card(self, name) :
 # return a card, that is show its contents
  current_card=name
  self.source.queue("suspense_music")
  self.source.play()
  ...

 def source_callback(self):
 # when the suspense sound is finished,
 # the card is returned
  if current_card!=None:
   self.cards[current_card].state='face'
```

Listing 2: The Style Sheet corresponding to the previous script

```
<blindstation>
 <scene name="memo">
  <widget name="card1">
   <state name="hidden">
    <rect x="20" y="20" w="40" h="80"/>
    <background file="..."/>
    <alt-text lang="en">Back of cards</alt-text>
    <alt-text lang="fr">Dos des cartes</alt-text>
    <shortcut key='A'/>
   </state>
   <state name="face">
    <background lang="en" file="HeartsJack.png"/>
    <background lang="fr" file="ValetCoeur.png"/>
    <alt-text lang="en">Jack of Hearts</alt-text>
    <alt-text lang="fr">Valet de Coeur</alt-text>
   </state>
  </widget>

  <sound name="intro_music">
   <file name="data/music.wav"/>
  </sound>
  <sound name="intro_speech">
   <file lang="fr" name="data/fr/intro.wav"/>
   <file lang="en" name="data/en/intro.wav"/>
  </sound>
 </scene>
</blindstation>
```

5.3 [Edutainment'2008]

Titre	**Towards Generalised Accessibility of Computer Games**
Auteur(s)	Dominique Archambault, Thomas Gaudy[(a)], Klaus Miesenberger[(b)], Stéphane Natkin[(a)] et Roland Ossmann[(b)]
Affiliations	[(a)]CNAM, Paris [(b)]Johannes Kepler Universität Linz, Autriche
Type	Conférence Internationale, avec Publication d'actes et Comité de lecture
Publication	Proceedings of Edutainment 2008 (3rd International Conference Technologies for E-Learning and Digital Entertainment), LNCS 5093, Springer, Berlin
Editeur(s)	Zhigeng Pan, Xiaopeng Zhang, Abdennour Rhalibi, Woontack Woo et Yi Li
Lieu/Date	Juin 2008, Nankin, Chine
Pages	518–527

Voir mémoire section **3.4.2**, page 42

Towards generalised accessibility of computer games

Dominique Archambault[1], Thomas Gaudy[2], Klaus Miesenberger[3], Stéphane Natkin[2], and Rolland Ossmann[3]

[1] Université Pierre et Marie Curie, INOVA/UFR 919, 9, quai Saint Bernard, 75252 Paris cedex 5, France,

[2] Centre de Recherche en Informatique du Cnam 292, rue St Martin, 75003 Paris, France

[3] Johannes Kepler Universität Linz, "Institut Integriert Studieren" Altenbergerstrasse 69, A-4040 Linz, Austria

Abstract. Computer games accessibility have initially been regarded as an area of minor importance as there were much more "serious" topics to focus on. Today, the society is slowly moving forward in the direction of accessibility and the conditions come to make new proposals for mainstream game accessibility. In this paper we'll show the main reasons why it is necessary to progress in this direction, then we'll explain how works standard computer applications accessibility and why it is not working in general with games. We will discuss the state of the art in this area and finally we will introduce our vision of future accessibility framework allowing games developer to design accessible games as well as assistive providers the possibility of developing Assistive Games Interfaces.

1 Introduction

Computer games have become an important part in child and youth culture, and most children, in developed countries, have considerable experience of such games. Additionally these games are used by a growing part of the population, including especially young adults (on average 25 years old, including 40% of women[1]) but the proportion of players is also growing in other age groups of the population.

A lot of people with impairment are excluded from the computer games world because of accessibility. Indeed games accessibility have initially been regarded as an area of minor importance as there were much more "serious" topics to focus on. Since the middle of the nineties, lots of works have been focusing on making office computer applications accessible, and it's a fact that nowadays word processor and spreadsheets applications are reasonably accessible as well as web browser and mail readers.

[1] TNS Sofres, *Le marché français des jeux vidéo* (The market of video games in France). afjv, November 2006.
`http://www.afjv.com/press0611/061122_marche_jeux_video_france.htm`

Today, as Zyda claims, *"the time has come to take computer games seriously, really seriously"* [1]. Indeed the mainstream commercial market for computer games and other multimedia products has shown an impressive growth in the last five years. Costs for the development of a game may reach the level of major movie productions, involving more than a hundred employees [2]. The expectation by games players of ever more impressive games has seen increasing development budgets, and with a more focused use of new technologies.

Academia and also R&D over the last years have started to focus on *"serious games"*. Leading experts speak of *"creating a science of games"* [1] with the goal of implementing games and game like interfaces of general importance for a growing number of applications and as a general trend in the design of Human-Computer Interfaces (HCI) [3].

In addition, general HCI is beginning to use concepts and methods derived from games as they promise an increased level of usability. Games and game-like interfaces are recognised as a means to implement educational, training, general HCI and web applications with usability and effectiveness. Particular examples of interest are:

- eLearning and edutainment which more and more implement or use didactic games [4]. As an example of this trend it should be noted that critical issues like mathematics and science education are approached with game-based learning infrastructures and edutainment to address the well known didactic problems in this domain. *"Games stimulate chemical changes in the brain that promote learning."* [5].
- Avatar based interfaces. We assist to a growing number of applications in such environments: for instance in France, real Job interviews have been organised in Second Life[2].
- Emerging Non Classical Interfaces (e.g. virtual/augmented reality, embedded systems, pervasive computing).
- A lot of Cultural Multimedia products, like Museum CD-Roms or DVD-Roms.
- Web 2.0
- Other software considered as inaccessible until today might come under accessibility discussions based on the principles, guidelines and tools developed for games and games like interfaces (e.g. simulation software, charts, virtual/augmented reality).

Then, even if it would be considered as questionable to use the limited resources that are available for research on accessibility to address problems of people with disabilities using games or edutainment software, the general evolution of HCI towards game-like interfaces compel us to also consider a "serious" look at games from the accessibility perspective, in order to keep pace with the general level of accessibility achieved over the last decades in standard HCI. When standard HCI changes also accessibility has to change and this is closely related to games.

[2] Linden Lab, `http://www.secondlife.com`

People with disabilities form one of those groups benefiting most from ICT[3]. Indeed, Assistive Technology (AT) enables them in a lot of situations in their daily lives, at school as well as at work or at home, in mobility, etc. The possibilities offered to the by eInclusion makes a difference in the life of a lot of people. Therefore it seems important that children get used to using technology as early as possible. Computer games are often a good training for the use of AT, for children as well as for adults after accidents, diseases. In addition playing games contributes considerably in establishing and ameliorating the skills in dealing with HCI.

From a different perspective, new approaches towards therapeutic and educational games for people with disabilities; children can benefit a lot from the use of computer games for their psycho motor and cognitive development [6].

We can now find a few hundreds of specific games, which have been developed especially for various groups of disabled users, but actually:

- this number is very short regarding to mainstream games, and they are often limited to one language,
- these games are usually very specifically dedicated to a extremely small group of end users with little or no access to the mainstream market (based on their abilities)
- these games are often very simple or old fashioned (even if a few very interesting exceptions exist)
- an important amount of these games are driven by specific pedagogical and therapeutic objectives and, on the whole, not much fun.

The limited budgets dedicated to specific developments make it very difficult to propose specific games with the quality and the size of mainstream games, which limits the possibilities of gaming experience for those players. Because of this, games for people with disabilities tend to worsen the segregation of disabled people from the mainstream gaming community they are the only games that they can interact with. This situation is in contradiction with the general eInclusive principles of ICT and Assistive Technology.

Accessibility of games is a more complex problem than software or web accessibility in general. The first reason, which seems obvious but is very important, is that: Accessible games must still be games! [7] Designing games that work for players with disabilities is quite a challenge: an important research, practical and social issue that has to be carried out now. This research should lead to one goal: the accessibility of mainstream games. Several aspects have to be taken into account: to find out how to handle game interaction situations with alternative devices, to develop models allowing to make mainstream games compatible with these alternative devices, to write according guidelines, methodologies and techniques.

To give people with disabilities the chance to have access to multimedia games should be seen as a great challenge for better eInclusion and participation in society.

[3] Information and Communication Technologies

The main groups of people addressed by these accessibility issues are those who cannot use mainstream games because their disability prevent them to use a modality which is necessary for some kind of games, namely:

– People who cannot use the ordinary graphical interface, because they are totally blind or because they have a severe visual impairment (sight rated < 0.05) [8] ;
– People who cannot use or have limited access to ordinary input devices like keyboard, mouse, joystick or game pad due to limited hand dexterity ;
– People with cognitive problems who need support to better understand the scene and react properly (e.g. symbol, text, speech and easy to understand support) ;
– People with hearing problems or deafness not able to accommodate to sound based interaction modalities ;
– People with problems in reacting to a strict time setting of the game out of various functional, cognitive and also psychological problems.

2 Software accessibility

Today it is state of the art that people with disabilities can interact with the standard desktop/WIMP[4] based interface using Assistive Technology. Specific access software applications, like screen readers and screen magnifiers, alternative input devices, alternative information rendering – sound, text, signs, colour/size/contrast of objects, allow them to access to many computer applications. This is mainly the case, as mentioned above, for text based software: word processors, spreadsheets, mail clients, web browsers. The problem is that these access software applications are not able to access any software application whatever the way it has been developed. Indeed they need to collect accessible information from the applications to render it using alternative output modalities, or to control them using alternative input modalities.

In other terms, to achieve accessibility of software applications, it is necessary to have accessibility support embedded in the applications. During the last decade, accessibility frameworks have been developed and are available in the main environments. For instance, Microsoft has developed Microsoft Active Accessibility[5], to make their Windows Applications accessible, application developers have to implement the IAccessible interface[6]. There exist similar frameworks on Mac[7] and on Linux desktop environments[8]. Theoretical works can be cited too [9]. Furthermore specific development frameworks need to support accessibility, for instance Java[9] and Mozilla[10].

4 WIMP: Windows/Menus/Icons/Pointers
5 MSAA: `http://msdn2.microsoft.com/en-us/library/ms697707.asp`
6 `http://msdn2.microsoft.com/en-us/library/accessibility.iaccessible.aspx`
7 Apple accessibility: `http://www.apple.com/accessibility`
8 Gnome Accessibility: `http://developer.gnome.org/projects/gap`
KDE Accessibility: `http://accessibility.kde.org`
9 Desktop Java Accessibility
10 Mozilla Accessibility Project: http://www.mozilla.org/access

It is not enough that applications respect accessibility standards. In most cases, content must be accessible too. For instance, in the case of a web site, the accessibility of web browser is necessary but the web contents must also be accessible. Graphical elements for instance must have textual alternatives, and this depends on the content itself. In that respect, the W3C launched the Web Accessibility Initiative to developed Web Content Accessibility Guidelines [10]. These guidelines indicate how to use each of the HTML tags to make a web site accessible. Accessibility of content has been developed in other content formats such as the proprietary PDF and Flash formats[11].

Of course there are still a lot of barriers in access to software and web content, but basically there are technical solutions asking for according political and practical measures to put this potential in place.

3 What is different in the case of games?

These accessibility solutions are working satisfactorily for standard desktop applications (WIMP based) but not for computer games. First the notion of working satisfactorily is a) not enough and b) not easy to define in that context.

Indeed, the results of a game can not be easily quantified, like in the standard case of classical desktop applications. In a word processing software, it is easy to measure the time needed by a user to write a document or to edit a document produced by a colleague. In a game we can as well observe if a player succeeds, and measure the time to finish a level or any case relevant for the game considered. But this is far not enough. Unlike others software, games have to provide special good feelings to players. There are probably some emotional factors to consider in the desktop applications, but they are usually not taken into account, or at least unless they affect the productivity. In the case of a game these factors are the most important.

Video images and audio messages contain emotional components and specific patterns which can easily be perceived and due to empathy can be experienced by the viewer and listener. For the same reason interactive video games are attractive and popular among the youth and adolescence. Empathic arousal has a strong influence on people viewing, listening and reading by forming their social response to the external events through mental estimation of the problem and simulation of possible solutions and actions [11, 12].

It has been demonstrated in numerous psychological studies that some emotions can motivate a specific human action and behaviour. The development of the emotional intelligence in youth depends on a social inclusion and personal experience which usually rely on observing others' actions and behaviours presented in a real life (the cultural *milieu*) and in the artificial situations disseminated by movies, television and video games [13–16]. Being deprived of access to information with emotionally rich content, blind and visually impaired children have experienced a significant emotional distress which can lead to depression and deceleration in cognitive development [17, 18].

[11] Adobe Accessibility Resource Center: `http://www.adobe.com/accessibility`

As we stated above: *Accessible games must still be games!* Visually impaired adults in work situation accept relatively big constraint on usability to be able to use the same software as their sighted workmates and to work on the same documents. This is not the case with children, especially playing. In other terms it is not enough to find a technical way allowing to access to all information needed in the interface, the result must be as interesting and as usable as the original game, and additionally it must be possible to succeed!

This helps us to understand that game interfaces are of a profound different nature than standard HCI and use their own technology (game engines). Usability and accessibility ask for freedom in time, speed, undo, mode of interaction,... It is a key criterion outlined by the W3C/WAI guidelines and software accessibility guidelines that the interface must not prescribe a certain interaction behaviour. But it is the core idea of games for realising immersion into a game, joy and setting up the gaming feeling to prescribe a restricted action and reaction behaviour and to force the user to be successful in this "reality". It seems that the more the player has to follow a strict behaviour, the more it seems that the game "takes the player into it" and puts immersion in place.

Therefore game accessibility goes beyond standard HCI and content accessibility measures. It must allow the prescription of behaviour by the system but it asks for alternatives and freedom of adaptation in the level of prescription and usage of modalities of interaction. If a mainstream game has put accessibility in place it is the role of adapted AT interfaces to realise immersion. Therefore it is inevitable to work on these adapted AT interfaces in game accessibility.

4 Game accessibility during the last decade

Even if a fair amount of games has been developed in this field during the last 5 years, today there are still very few games that are accessible (including specific games and mainstream games). Coming back to the year 2000, one could only find a very short number of games usable by disabled players.

4.1 Specific games

The first period that we identified is the period 2000-2005, that we will call the "basic studies". During this period we have seen the development of various games specifically designed for specific groups of people with disabilities. These kind of games are usually funded by foundations or non-profit organisations. Most of them are very nice for the group they were developed for but have little interest for mainstream, except maybe a few of the audio games. What is additionally of importance in the context of this proposal is that they demonstrate how to render various interaction situations with alternative modalities. This can be supplemented by the number of research papers about new uses of various modalities in the game play (Braille devices, haptic...).

The largest number of such games concerns audio games. Actually audio games include three different concepts in which the main play modality is audio.

The first meaning involve mainstream video rhythm games like Guitar Hero II. The second definition is related to artistic musical experiments. The third correspond to games which is based on sound environment (sound scenes, sound characters, actions) and can be played without vision, and therefore are accessible to visually impaired players (like interactive audio books, stories and tales): In 10 years, over 400 accessible audio games have been developed (which is very small as compared to video games).

The web site `http://audiogame.net` refers interesting interactive works. There exist a few visual audio games which can be very impressive and playable as well with or without sight. Terraformers [19] was developed with accessibility as part of the original concept of the game. On the other hand, AudioQuake [20] was developed as a research project to make a non-accessible game accessible.

A few tactile games can be found, that are games where the inputs and/or the outputs are done by tactile boards or by Braille displays, in combination with usually audio feedback. The use of Braille displays for gaming is only experimental by now. Some research is currently carried out in order to find models to represent a 2D space on a linear Braille display [21]. A few experimental games were designed in order to evaluate these models, for instance a snake game and a maze game. During the TiM project [22] (IST-2000-25298), a number of tactile games have been created or adapted from existing mainstream contents. Tomteboda resource centre in Sweden have published a report where they try to stimulate parents and educators to develop their own games using tactile boards [23].

[24] studied the possibilities offered by haptic technologies for creating new interactions usable by blind people. He worked especially with the Senseable Phantom. Then a number of papers explore the possibilities of using haptics in experimental games: [25–30]...

The outcomes of this period is that we can base our work now on a lot of studies on playing games in various situations of functional limitation and on the adaptation of computer game situations.

4.2 Setting up the foundations

The second period, ongoing since 2005, sees the emergence of the notion of games that work for all. This is declined in 2 aspects: game designed for all and accessibility of mainstream games. It is driven by the already mentioned fact that games and game like interfaces are recognised as important contributions to the next generation of HCI, eLearning and other applications.

The goal of games being developed under the title of designed for all is to give players with all different kinds of abilities or disabilities the opportunity to play these games. This requires a very advanced game setting and configuration. UA-Chess [31] is a universally accessible Internet-based chess game that can be concurrently played by two gamers with different (dis)abilities, using a variety of alternative input/output modalities and techniques in any combination. Access Invaders [32] is a designed for all implementation of the famous computer game Space Invaders, with the target groups of people with hand-motor impairments, blind people, people with deteriorated vision, people with mild memory/cognitive impairments and novice players. The approach of [33] was to make an already published game accessible and demonstrate the feasibility and the effort necessary to fulfil this goal. It is based on an open-source implementation of Tic-Tac-Toe.

Games designed for all must be seen as examples of good practise, demonstrating that Universal Access is a challenge and not utopia. In this projects we have to admit that the various alternative access features to these games require more development than the rest of the game itself.

Following these experiments it became clear that the accessibility of mainstream computer games needed to be improved. [7] proposes a set of rules to make computer games accessible by visually impaired people, derived from the TiM project. We started to work on formulating Guidelines for the Development of Accessible Computer Games, covering a wide range of disability groups [34]. IGDA[12] published a white paper on Accessibility [35], showing early signs of interest from the mainstream gaming industry.

5 What is needed now?

We have developed the reasons why computer game accessibility should be considered seriously. Then we have seen how accessibility works in the case of standard desktop applications and why current accessibility frameworks it would not work with games or game-like applications. In the previous section we have seen that a lot of works have been studying how to render different game situations using different kinds of alternative devices.

Players with disabilities need to use Assistive Technology to play accessible games. But contrarily to any other computer application, this must not take off the characteristics of these applications that make them games. It is not only the task which one fulfils with an application (e.g. with office/mail software) but it is the procedure of playing the game it self which is fun and which provides learning benefits. In other terms, games accessed with AT still must be games and due to this it challenges the usage of AT. Then increasing accessibility of games will mean developing a new generation of assistive software taking into account much more parameters than current AT has access to, via the existing accessibility frameworks: characterisation of information available (including ranking of importance regarding the current task to fulfil), relative importance of events, speed, etc.

These new assistive software applications, which we will call *Assistive Game Interfaces* (AGI) will not likely be unique for a specific kind of impairment (like today a screen reader allows to access any office application). Depending of the ability constraints, some could be dedicated to a specific game or a game engine, some would be dedicated to a kind of games and finally some others would be generic (covering a large range of games).

[12] International Game Developers Association

We could for instance imagine a "captioning application" allowing lots of different games to have captions when a character is speaking. On the other hand, for blind gamers, we could have specific AGI for text based games, another AGI working with a specific game engine, and a third one dedicated to a popular car race (since in this case the interaction would have to be completely redesigned).

To achieve these goals, the AGI will need to collect information from the core of the game itself. Indeed most of the information needed cannot be efficiently processed automatically from the mainstream game (for instance the captioning information). We have seen that the existing accessibility frameworks are not sufficient to provide these AGI with the necessary information. This means that it is necessary to design a new Game Accessibility Framework (GAF). This framework will have to take into account the specific data needed by various alternative devices to work properly. To continue with the example of the Captioning application, this application will need access to the complete transcription of the texts spoken by the characters in the game. The Game Accessibility Framework will have to specify how and in what format.

The first steps to carry out the specification of the Game Accessibility Framework are (a) a typology of game interaction situations and (b) a characterisation of Accessibility in terms of functional requirements. From the study of these expected results, the specification of the GAF can be produced, including the data formats and exchange protocols to transmit information between game and AGI.

Now it is time to make a significant move. This implies some participation from assistive technology specialists as well as from mainstream games developers.

The proposed solution may seem not realistic but it has to be considered that:

- The state of the art shows that the technology is ready
- This solution will be the lighter for game developers (consider for instance the work that would be needed to add a "caption" option in a game, compared to the implementation of the access to texts that are already existing somewhere in the production process).
- the societal need for improving inclusion is growing in some leading countries (Northern European countries, Austria, Canada, Japan, etc) and the political pressure will necessarily follow, leading to laws and recommendations. We expect that this situation extends to the rest of Europe and North America, and to the rest of the world.
- the evolution of standard HCI towards game like interfaces will soon make these applications enter in the scope of existing laws

References

1. Zyda, M.: Creating a science of games. ACM Communications **50**(7) (jul 2007)

2. Natkin, S.: Video games and Interactive Media: A Glimpse at New Digital Entertainment. AK Peters (2006)
3. Kellogg, W., Ellis, J.and Thomas, J.: Towards supple enterprises: Learning from N64's Super Mario 64, Wii Bowling, and a Corporate Second Life. In: "Supple Interfaces": Designing and evaluating for richer human connections and experiences. (sep 2007)
4. Chatham, R.E.: Games for training. ACM Communications **50**(7) (jul 2007)
5. Mayo, M.: Games for science and engineering education. ACM Communications **50**(7) (jul 2007)
6. Hildén, A., Svensson, H.: Can All Young Disabled Children Play at the Computer. In Miesenberger, K., Klaus, J., Zagler, W., eds.: Proc. ICCHP 2002 (International Conference on Computers Helping People with Special Needs). Volume 2398 of LNCS., Linz, Austria, Springer (July 2002)
7. Archambault, D., Olivier, D., Svensson, H.: Computer games that work for visually impaired children. In Stephanidis, C., ed.: Proceedings of HCI International 2005 Conference (11^{th} International Conference on Human-Computer Interaction), Las Vegas, Nevada (July 2005) 8 pages (proceedings on CD-Rom).
8. Buaud, A., Svensson, H., Archambault, D., Burger, D.: Multimedia games for visually impaired children. In Miesenberger, K., Klaus, J., Zagler, W., eds.: Proceedings of ICCHP 2002 (International Conference on Computers Helping People with Special Needs). Volume 2398 of LNCS., Linz, Austria, Springer (July 2002) 173–180
9. van Hees, K., Engelen, J.: Non-visual access to guis: Leveraging abstract user interfaces. In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I., eds.: Proc. ICCHP 2006 (10^{th} International Conference on Computers Helping People with Special Needs). Volume 4061 of LNCS., Linz, Austria, Springer (July 2006) 1063–1070
10. W3C: Web Accessibility Initiative — Web Content Accessibility Guidelines 1.0. Technical report, World Wide Web Consortium (W3C) (May 1999) [http://www.w3.org/TR/WAI-WEBCONTENT].
11. Grézes, J., Decety, J.: Functional anatomy of execution, mental simulation, observation, and verb generation of actions: a meta-analysis. Human Brain Mapping 12 (2001) 1–19
12. Prinz, W., Meltzoff, A.: An introduction to the imitative mind and brain. In Meltzoff, A., Prinz, W., eds.: The imitative mind: Development, evolution and brain bases. University Press, Cambridge (2002) 1–15
13. Segall, M.H., Campbell, D.T., Herskovits, M.J.: The influence of culture on visual perception. Studies in Art Education **10**(1) (1968) 68–71
14. Patterson, J.: Theoretical secrets for intelligent software. Theory into Practice **22**(4) (1983) 267–271
15. Fromme, J.: Computer Games as a Part of Children's Culture. The International Journal of Computer Game Research **3**(1) (May 2003)
16. Sebanz, N., Knoblich, G., Prinz, W.: Representing others' actions: just like one's own? Cognition **88** (2003) B11–B21
17. Barresi, J., Moore, C.: Intentional relations and social understanding. Behavioral & Brain Sciences, 19(1) (1996) 107–122
18. Chartrand, T.L., Bargh, J.A.: The chameleon effect: the perception-behavior link and social interaction. Journal of Personality and Social Psychology 76 (1999) 893–910

19. Westin, T.: Game accessibility case study: Terraformers - a real-time 3d graphic game. In: Proceedings of the Fifth International Conference on Disability, Virtual Reality and Associated Technologies, Oxford, UK (2004) 95–100
20. Atkinson, M.T., Gucukoglu, S., Machin, C.H.C., Lawrence, A.E.: Making the mainstream accessible: What's in a game? In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I., eds.: Proc. ICCHP 2006 (10^{th} International Conference on Computers Helping People with Special Needs). Volume 4061 of LNCS., Linz, Austria, Springer (July 2006) 380–387
21. Sepchat, A., Monmarché, N., Slimane, M., Archambault, D.: Semi automatic generator of tactile video games for visually impaired children. In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I., eds.: Proc. ICCHP 2006 (10^{th} International Conference on Computers Helping People with Special Needs). Volume 4061 of LNCS., Linz, Austria, Springer (July 2006) 372–379
22. Archambault, D.: The TiM Project: Overview of Results. In Miesenberger, K., Klaus, J., Zagler, W., Burger, D., eds.: Proc. ICCHP 2004 (9^{th} International Conference on Computers Helping People with Special Needs). Volume 3118 of LNCS., Berlin, Springer (July 2004) 248–256
23. Hammarlund, J.: Computer play for children who are severelly visually impaired: Using an alternative keyboard with tactile overlays. Technical Report 20, Tomteboda resource centre, Stockholm, Sweden (1999)
24. Sjöström, C.: The sense of touch provides new computer interaction techniques for disabled people. Technology and Disability **10**(1) (1999) 45–52 IOS Press, Amsterdam, The Netherlands.
25. Johansson, A.J., Linde, J.: Using Simple Force Feedback Mechanisms as Haptic Visualization Tools. In: Proc. of the 16th IEEE Instrumentation and Measurement Technology Conference. (1999)
26. Wang, Q., Levesque, V., Pasquero, J., Hayward, V.: A Haptic Memory Game using the STRESS2 Tactile Display. In: CHI2006, Montréal, Québec, Canada, ACM (April 2006)
27. Raisamo, R., Patomäki, S., Hasu, M., Pasto, V.: Design and evaluation of a tactile memory game for visually impaired children. Interacting with Computers **19**(2) (2007) 196–205
28. Evreinov, G., Evreinova, T., Raisamo, R.: Mobile games for training tactile perception. In: ICEC 2004 – Third International Conference on Entertainment Computing. Volume 3166 of LNCS., Eindhoven, The Netherlands, Springer (September 2004) 468–475
29. Crossan, A., Brewster, S.: Two-handed navigation in a haptic virtual environment. In: Extended Abstracts of CHI2006, Montréal, Québec, Canada, ACM (April 2006)
30. Rodet, X., Lambert, J.P., Cahen, R., Gaudy, T., Guedy, F., Gosselin, F., Mobuchon, P.: Study of haptic and visual interaction for sound and music control in the Phase project. In: Proceedings of the 2005 conference on New interfaces for musical expression, Vancourer (May 2005) 109–114
31. Grammenos, D., Savidis, A., Stephanidis, C.: Ua-chess: A universally accessible board game. In Salvendy, G., ed.: Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction, Las Vegas, Nevada (July 2005)
32. Grammenos, D., Savidis, A., Georgalis, Y., Stephanidis, C.: Access invaders: Developing a universally accessible action game. In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I., eds.: Proc. ICCHP 2006 (10^{th} International Conference on Computers Helping People with Special Needs). Volume 4061 of LNCS., Linz, Austria, Springer (July 2006) 388–395

33. Ossmann, R., Archambault, D., Miesenberger, K.: Computer game accessibility: From specific games to accessible games. In Mehdi, Q., Mtenzi, F., Duggan, B., McAtamney, H., eds.: Proceedings of CGAMES'06 Conference (9th International Conference on Computer Games), Dublin, Ireland (November 2006) 104–108
34. Tollefsen, M., Flyen, A.: Internet and accessible entertainment. In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I., eds.: Proc. ICCHP 2006 (10^{th} International Conference on Computers Helping People with Special Needs). Volume 4061 of LNCS., Linz, Austria, Springer (July 2006) 396–402
35. Association, I.G.D.: Accessibility in games: Motivations and approaches (2004) [http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf].