An ideal Braille transcriber ?
Issue, modeling, conception and implementation of a prototype

B. Mascret*, A. Mille*, M. Ollier**

* Lyon Research Center for Images and Intelligent Information Systems (LIRIS), UMR 5205, team Silex (Supporting Interactions and Learning by Experience), Claude Bernard Lyon 1 University (France), bruno.mascret@liris.cnrs.fr
** National high institute of training and research young disabled's education and adapted learning (INS HEA), Suresnes (France)

Abstract

Even though a great effort has been made to inform institutions and the general public, having access to digital information is still nowadays a real handicap for visually impaired people. Transcribing heterogeneous contents (scientific idioms, literature, full or contracted braille, music...) is one of the major considerations limiting a braille reader's accessibility to digital documents. Indeed, strict conventions rule differently each type of content, which therefore must be previously and correctly identified. After a quick introduction to the braille transcription issue, this article will develop the different stages in creating an "ideal" transcriber : modeling, conceiving and implementing. This transcriber's main goals are to ease the transcription chain, to be fast enough for a real-time use, and to propose a multiple modal tool, therefore independent of any operating system and providing a polymorphic use.

Key words

Accessibility, assistive technologies, Braille, adaptation of documents, transcription, transcriber.

Résumé

L'accès à l'information numérique des déficients visuels, malgré les réels efforts de sensibilisation entrepris, reste encore une source de difficultés. Un des principaux facteurs limitant l'accessibilité d'un document numérique aux utilisateurs de Braille provient de la complexité à transcrire des documents composites (expressions scientifiques, brailles littéraire et abrégé, musique...), chaque contenu utilisant des normes particulières et devant être préalablement identifié. Cet article introduit la problématique de la transcription du braille, puis détaille la modélisation, la conception et la mise en œuvre d'un transcripteur « idéal » : simplifiant la chaîne de transcription, multimodal, indépendant des systèmes d'exploitation, polymorphe, et suffisamment rapide pour le traitement en temps réel. Nous concluons en présentant les limites de notre solution et en proposant de nouvelles pistes sur la manière d'assister l'utilisateur à partir de l'expérience tracée.

Resumen

El acceso de los deficientes visuales a la información numérica, a pesar de los notables esfuerzos de sensibilización que se emprendieron, sigue constituyendo todavía una fuente de dificultades. Uno de los principales factores que limitan la accesibilidad de un documento numérico a los usuarios de braille proviene de la complejidad que supone transcribir documentos compuestos (expresiones científicas, brailles literarios y estenotipia, música...). Cada tipo de contenido utiliza normas particulares que deben identificarse previamente. Este artículo introduce la problemática de la transcripción del braille, luego detalla la modelización, la concepción y el desarrollo de un transcriptor "ideal" capaz de simplificar la cadena de transcripción, siendo multimodal, independiente de los sistemas operativos, polimorfo y lo suficientemente rápido para el tratamiento en tiempo real. Concluimos presentando los límites que conlleva nuestra solución y proponiendo nuevas pistas sobre la manera de prestar asistencia al usuario a partir de la experiencia trazada.

1. Introduction

Second generation technologies are increasingly making digital data access easier, especially considering the use of CMS, of educational platforms and many other communication tools. The amount of online information grows steadily in all fields -school, industry, business. Therefore, the CMS is a very precious publication tool for many. However, this context causes new activity limitations for visually impaired people -mainly extra work and a great time loss. The first part of our article will refer to Uzan's work - introducing the notion of "time disability" - to show to what extent this situation can be prejudicial to the visually impaired [8][9].

We will then focus more specifically on Braille users, with a quick presentation of the main issues linked to Braille and the existing transcribing tools. We will show how the transcriber NAT[1] aims at minimizing activity limitations and proposing a new approach of included automatic transcription. We will finally try to answer our question : "Can we consider an ideal Braille transcriber ?" after analyzing the limitations of our solution and introducing new leads, such as the trace-based reasoning.

2. Issues in relation with Braille and digital documents

We will now let the reader become familiar with the environment of handicap and visual impairment. We insist on the difference between impairment - an objective, physical, neurobiological or cerebral characteristic, most of the time permanent - and an activity limitation, linked to a particular context and likely to disappear if the environment changes. An "activity limitation" is defined by a beginning, a time period and possibly an ending. As an example, you would be in a activity limitation if you had to read an article written in Chinese and didn't know the language ; and this until the article were translated (or if you found a translator, learned Chinese etc.).

2.1. Introduction to the visually impaired persons' issue

2.1.1. Blind users and computers

Digital communication tools offer many advantages to most users : time-saving on diffusion, organization, less paper use... But they create new handicap situations to visually impaired persons. The accessibility of these communications media is increasingly better taken into account, thanks to specific software adaptations for visual impairment and to the development of "design for all". However it is mainly based on restoring the structure, contents and conditioning of websites or applications. Reducing the time needed to find a piece of specific information inside a document [9] [10], especially in a particular context, remains complicated and difficult.

Blind people only have sequential access to information (vocal reading, refreshable Braille display) and therefore cannot directly find the interesting elements, nor consider a document as a whole. The accessibility of digital contents by the document's producer is only rarely considered. Indeed a Braille user needs an adaptation processing to read digital documents containing scientific notations. A vocal synthesizer or a screen reader can give a general idea of the document, but they do not offer an accurate understanding. Besides, they are still unable to transcribe mathematical, musical or graphical notations.

2.1.2. Braille printing production

Printing a document in Braille (embossing) requires beforehand preparation : page setting, transcribing into 6-dot Braille, coding, etc (see ahead). Using specific adaptation software for Braille increases the data processing time, and often the price of the equipment (already very expensive). Moreover, all these stages make a very complex transcribing chain. Here is the example of a process leading to a proper embossing of a document - containing mathematical equations and word processing : first convert the mathematical notations into MathML using mathtype ; save the word document into a text format ; transcribe the mathematical contents using BraMaNet and save the result ; treat the text using Duxbury ; emboss.

A blind person needs four different softwares and handles at least three different files to produce the paper document, whereas a sighted person only needs a mathematical software and one file. Furthermore, the Braille printing producer is forced to use expensive software : mathtype 6 (120 €), microsoft office 2007 (150 €), Duxbury 10.6 (465 €), Windows vista (100 €). Considering BraMaNet is free, the expense is worth around 800 € (about 1100 $). The cost of the total equipment can seem insignificant for regular transcribers, but it may discourage occasional ones.

2.2. Introduction to the Braille issue

Transcribing into Braille is not, as one should think, just a plain transcoding of characters. The Braille alphabet is composed of 6-dot cells, as shown below (fig.1). A document used by a sighted person is called a black document as opposed to a Braille document.

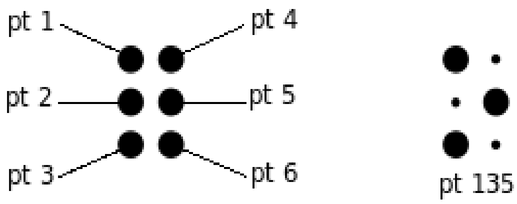

Figure 1. A Braille cell, 3x2 matrix, and each dot's name (1 to 3 on the left and 4 to 6 on the right). The code dots 1-3-5 shows the points 1, 3 and 5 raised, which codes for letter "o" in French.

Computer Braille (or 8-dot Braille) is only used with Braille displays or terminals. It has 2 more dots (matrix 4x2) added respectively on the bottom of each column (dots 7-8). This Braille is only used with computers in order to wipe out ambiguities created when the displayer transcodes bijectively instead of transcribing, and is thus never used on a paper base. For example, the capital letter A should usually be transcribed by two Braille characters : in French dots 4-6 (capital letter prefix) and dot 1 (letter a). But with the 8-dot Braille, we use dot 7 as the capital letter indicator ; therefore only one character is needed (dots 1-7). The 8-dot Braille offers much more combinations than the 6-dot Braille (256 instead of 64). However some users read it slowly because characters are more difficult to distinguish, and many prefer the 6-dot Braille.

Braille users are mainly blind persons and transcribers. However, a transcriber is not necessarily professional - for example, a teacher with a blind student in his class needs to transcribe his documents into Braille. Many countries, such as France, tend to integrate most blind students into general courses.

Different types of Braille codes are mentioned : the literary code (transcribing each character of the initial document), the abbreviated code (reducing the number of characters thanks to complex contraction rules), the mathematical code (transcribing all scientific notations)[3], the musical code. Each language has its own number of characters (no accents in English but many sorts of accents in other languages). Therefore, each language has its own Braille codes[2], if not each country (British and American codes are different). The musical Braille code is the only universal one. We sometimes use transcribing standards, describing page settings for documents, tables, notes, etc.

Braille code table	Representation of ⠲ (dots 2-5-6)	Representation of ⠄ (dot 3)
TBFr2007	. (dot)	' (apostrophe)
CBISF	/ (slash)	. (dot)

Figure 2. Table showing that differences between TBFr2007 and CBISF standards for character encodings of . and ' may lead to confusions when reading or transcribing (as for dot 3 in this example).

Finally, each country has its own Braille code tables associating black characters with Braille characters. Braille printers, terminals and fonts use a representation mechanism (and not a transcribing mechanism). Therefore, encoding a file is an important step leading to a good representation of the transcription. If the file encoding is compliant (ISO 8859 1, UTF 8, etc.), Braille code tables allow a satisfying display on tactile devices or during Braille printing.

2.3. The transcribing issue

Transcription standards are elaborate and precise enough for a human transcription to be good quality. On another hand, automatic transcriptions still have many problems, such as switching over from two-dimensional notation to linear text (mathematical idioms)[3], using the context to wipe out ambiguities, or handling complex abbreviation rules...

Of course, a professional transcriber can read, analyze and correct an automatically transcribed text. Many occasional transcribers are unable to do so, if they even know that the result is not perfect. A fairly simple document will be possible to read despite a few errors. However, it will be very difficult or even impossible to read if it contains complex and mixed contents (tables, pictures, formulas). Transcription centers exist and propose a high-quality job. However, they are very slow considering the communication and transcription delay. For all these reasons, a good-quality and automatic transcription would avoid time loss.

Visually impaired persons already have extra work to manage. Therefore the transcribing chain must be short and all tools have to be accessible. An automatic transcriber must be efficient, simple for an occasional user, accessible to blind persons, and customizable for master transcribers looking after extra time.

3. Critics of existing and automatic transcribing solutions

We will only propose an overall critic of ongoing solutions, not a detailed state of the art of French automatic transcription (see [7]). Their main drawbacks are usually the operating chain being too complex, and the time-lag between sighted and blind people accessing digital data with comparable quality. Several softwares propose specialized online library services. However the transcription is not a real-time one and specific material configurations are needed.

It is important to consider material or/and software solutions required for a good-quality transcription. If it is too expensive, it may exclude certain users. As for the user interfaces, most solutions only present one possibility : either through the web, through a software, in text mode or in graphic mode. Some of them aren't very ergonomic. They are mainly based on dictionaries which are sometimes assisted with transformation rules. Therefore updating them to French-speaking Braille evolutions is complicated.

4. Going towards an "ideal" transcriber

NAT was created in 2005 during a university project. It continued developing for almost two years without financing. In July 2007, the software received a first financial support from the European Social Fund and since July 2008 it is entirely supported by the French Minister of Education. The LIRIS French laboratory now supervises the project. An expert partnership with the INS HEA has been made in order to validate the quality of transcriptions.

This project mainly aims at solving the problems previously described, and wishes to produce a solution accessible to every one, independent from special configurations, polymorphically used, and potentially integrated to other systems. The motive is not to compete with transcribing centers - they are far better than any automatic software could ever be -, but on the opposite to give them a tool allowing a bigger efficiency and productivity.

4.1. General structure.

Taking all constraints into account has led us to a modular organization, based on adaptation to each type of document (format, mixed contents, encodings, etc.). The structure proposes three main modules: conversion, transcription and post-processing. Ideally the user gives the system a file in a given format : the conversion module conforms to the document type and produces an internal format file. Then the transcription module transcribes the internal file with chosen filters. Finally the post-processing module manages the presentation, exportation or printing. The specific role of each module allows the system to be independent during the development process. A new format would only need to be associated to a specific converter. Transcribing filters are independent from the initial format as well.

The transcription mechanism is original because the filters and their specialization are interoperable. Their implementation is no longer based on dictionaries but on rules, and therefore gets as close as possible to a human reasoning when using different transcribing processes. Since these filters are interoperable, they allow each document to realize dynamically its own transcribing scenario : using abbreviated or literate Braille code, choosing encodings, choosing the Braille code tables, whether or not transcribe mathematics.

4.2. Technical choices

This type of organization is based on using an internal file format to operate the transcription. This format must provide a degree of granularity sharp enough to analyze the phrases it will transcribe, but at the same time it must remain flexible enough to allow progress if new types of contents (music for example) were to be added. Likewise, programming languages should be executed on different platforms and modes (through web, interfaced application or service application).

We chose a solution based on java and API JAXP, which makes the processing of XML files and XSL filters for documents' transformations easier. The internal format has a fairly simple basis and allows other notation standards (such as MathML) to be integrated. The smallest element is the word or the punctuation mark. All types of contents (mathematical or literary) are organized inside paragraphs which constitute the document. For the time being, the different elements' properties are represented through tag attributes, and not as being themselves tags (this is different from HTML). This way the document maintains a simple structure.

Using open and standard formats as entry (open document, simple text file, MathML) guarantees NAT to be compatible on upfront with all softwares respecting international standards and independent towards specific software distributions. For example, we can use Open Office, StarOffice, KOffice ... to produce an ODF document.

4.3. Rules implementation through XSL style sheets

Each type of content has its own transcribing difficulties. A literary text is composed of many words, but its structure is fairly simple. A mathematical text has far less words, but its structure is more complex. The data processing we associate to each type of content is thus very diverse. The literary and the abbreviated Braille codes require a great deal of chain manipulation, whereas the mathematical and music codes require more nodal rules.

4.3.1. Transcribing a literary Braille code text

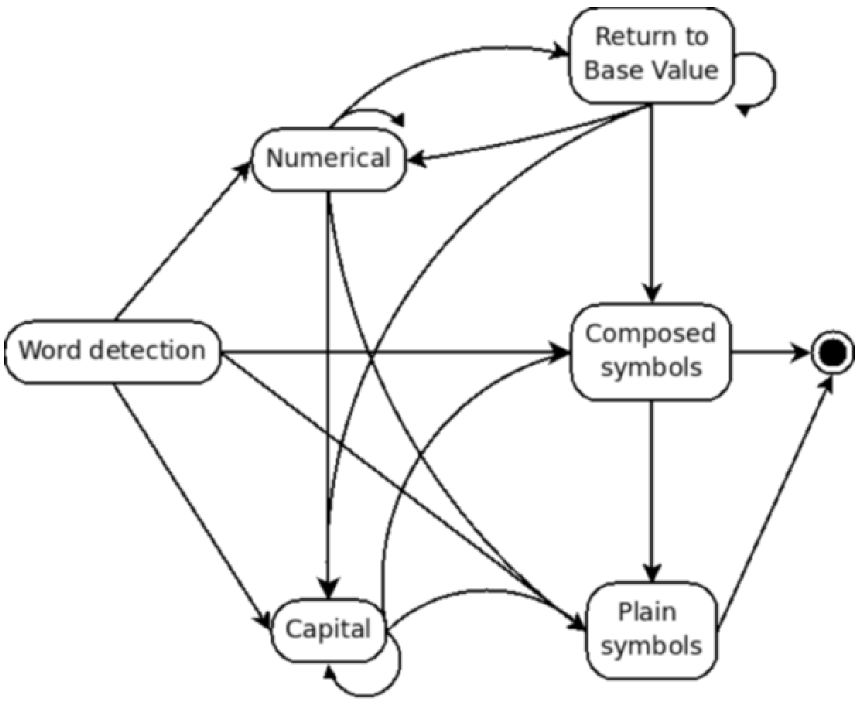

Figure 3. UML simplified state machine diagram of the transcription of a word into literary Braille. We use five main templates : Capital letter, Numerical, Return to base value (RBV), composed symbols and plain symbols. Three of them can call on to themselves (Numerical, Capital letter and RBV), and two of them can directly produce a final state (composed symbols and plain symbols).

The transformation sheet algorithm for the French literary Braille code is composed of several templates, and is mainly based on recursion. The aim is to produce as fast as possible a final state (transcription of a word). From the start, the templates are called on a precise order, which determines the state into which they transmit the word to the next template (during the transcribing process). Some templates need to determine the context into which they will act (for example, active capital letter prefixes), and to know the nature of the preceding and following elements. This organization is represented in the diagram in figure 3.

Transcribing the French word "Bonjour" uses the Capital letter template (because of the B), and then directly uses the plain symbols template. Here is the result :

⠨⠃⠕⠝⠚⠕⠥⠗

The first code (dots 4-6) is the capital letter prefix, the other cells match to "bonjour".

On the other hand, the word "88Haüy"'s path will be : numerical, RVB (because in French ü has the same representation as 8), capital letter, plain symbols.

This leads to the representation : ⠠⠳⠳⠰⠨⠓⠁⠳⠽

Dot 6 indicates that the word contains numbers. "88" is transcribed by dots 1-2-5-6 dots 1-2-5-6. Then the code dot 5-6 indicates that we are no longer in a numerical context, and back to the basic values. Dots 4-6 is the capital letter prefix. Dots 1-2-5 is "h", dot 1 is "a", dots 1-2-5-6 is "ü" (and no longer "8"), and dots 1-3-4-5-6 is "y".

```
- Number prefix rules
----------------------
<digit, maintain_NP;      ; digit  ; ""          >
<math_debut       ;       ; digit  ; ""          >
<digit,letters     ;      ; math_debut digit; "" >
<digit,maintain_NP,letters; ; math_debut digit;"">
<            ;    ; math_debut digit; NP_Braille>
<            ;    ; digit; NP_Braille; "" >
[etc...]
- Capital prefix rules
<(CAPITALPREFIX=CAPITALPREFIXON)upperletter;      ;      upperletters,
lowerletter;  >
<(CAPITALPREFIX=CAPITALPREFIXON)
[etc...]
-- Symbol and character rules
[etc...]
<   ; "ü"    ;    ; p1256 >
```

Figure 4. Winbraille transformation rules for "ü" and capital letters. No going backwards is possible because of their order. In our example, the "ü" will be incorrectly transcribed because the "numerical" context has been lost, and we will read "88ha8y".

This organization has two main advantages. First it is possible to combine different rules in a dynamic way (because each template can be called on at any time of the process, and not just at one precise moment). Secondly it transcribes different words faster because it only calls the necessary templates for this particular chain. Winbraille's "dictionary" rule files are planned to activate different reading contexts (capital letters, numerical contents, etc.). However each rule sequence must have been foreseen beforehand in the rule files and generates new iterative rules every time. As the number of simultaneous rules grows, the number of represented combinations becomes exponential.

4.3.2. Mathematics transformation sheet

The processing system here is different : the mathematical vocabulary is fairly simple, but the structures are very complex. The MathML format (recommended by w3c) represents these structures rather accurately and identifies each vocabulary element with specific tags (depending on its content's role) : mo (math operator), mi (math identifier), mn (math number), mtext (math text)..., which makes it easier to discriminate each symbol. MathML opens two encoding possibilities : content MathML (used to represent an expression's logical structure), and presentation MathML (used to graphically represent a mathematical notation). Content MathML would ease the transcribing process, but unfortunately it is not used within general public softwares because they mainly wish to have a proper graphic rendering. Transforming content MathML into presentation MathML is easier than the opposite therefore we have chosen presentation MathML for NAT's internal format in mathematical representation.

Rather than having a great deal of translation dictionaries using several entry formats (LaTeX, MathML, etc.) and adding a transcriber to each format - which is the case of the Vickie mathematical transcriber[6] -, NAT integrates directly MathML in the internal format. Realizing XSL transformation sheets - able to adapt to the context while transcribing an idiom - now becomes possible, as we did for the literary Braille code. NAT uses and slightly improves the MathML transcribing techniques (through XSL) used within BraMaNet. New MathML entities have been implemented and the general sheet organization has been improved.

The Vickie project uses a dictionary associating an only representation for each element and coming at the end of the process (reading module). It is rather efficient, but still doesn't adapt to the context. For example, a prefix indicating a code change usually precedes a mathematical content (dot 6 & dot 3). However, if the content is simple - when using bloc symbols is not necessary -, only the mathematical modifier needs to be activated (dot 6). This way dot 3 precedes the fraction one half in case (1), but in case (2) the indicator is dot 6 dot 3 (because "x + 1" must be placed between Braille blocs).

$$\frac{1}{2}+3 \quad (1) \qquad\qquad \frac{1}{2}*\frac{3}{x+1} \quad (2)$$

Therefore in Braille we cannot associate a unique value to the root tag of an equation. In some cases entities can have different meanings (depending on the context). For example, the "→" (right arrow) entity can be either a simple arrow or a vector. We are sure that the content MathML would be the ideal solution to wipe out these ambiguities. However it remains a minority implementation. The XSL code shown in figure 5 illustrates the way NAT tells these two possibilities apart. At the moment we are working on a solution to directly taking into account content MathML, without converting it into presentation MathML.

```
<xsl:when test=".='&rarr;' or .='&xrarr;'">
 <xsl:choose>
  <xsl:when test="local-name(parent::*)='mover'">
   <xsl:text>&pt46;&pt25;</xsl:text><!--vector-->
  </xsl:when>
  <xsl:otherwise>
   <xsl:text>&pt456;&pt156;</xsl:text><!--arrow-->
  </xsl:otherwise>
 </xsl:choose>
</xsl:when>
```

Figure 5. Piece of the NAT XSL sheet, where the distinction between the right arrow and the vector - represented in the same way using MathML 1, is made.

5. Limits and future prospects

NAT's latest version handles entirely the literary and mathematical Braille codes. Recently, a syndicate of experts, developers and testers has been established. It should improve the software's quality and shortly include transcribing functionalities for abbreviated and musical Braille codes. Despite all this, can we say it is a perfect transcriber? Several cases still remain ambiguous and problematic. The literary Braille code is fairly simple to deal with, but the abbreviated code will set many questions that only human intelligence is yet able to solve without mistake. The end of our article will bring to light the main factors limiting an automatic transcription's quality and will propose new ways of considering assistive technologies through traced-based reasoning and experience.

5.1. The "ideal" automatic transcriber's limitations

In the field of accessibility to digital data, one of the aspects we usually ignore is the time needed to access and get acquainted with a piece of information [8] [9]. The problem is restrained thanks to synopsis, internal links and other navigation tools. However they do not always allow a blind person to interact with a digital document as a sighted person would. Information remains sequential and brings on a considerable time loss. Using paper on exit can avoid the linear representation of computer Braille. It also opens on two advanced possibilities of information space positioning (for example representation of matrix, tables, ergonomic page settings, etc). However we also lose many interaction possibilities to the document (corrections, researches, notes...). Many contents remain almost impossible to transcribe with software (pictures, illustrations, graphics...).

We do not pretend we can realize a perfect transcriber, but we describe NAT as an adaptable tool which can ease the transcribing process and offer many users functionalities allowing time gain. In this way, professional transcribers use it as a working basis whereas people who don't know Braille can produce usable documents for blind. NAT's transcriptions are thus only a first step leading to a document's accessibility. They are in no means a final result.

5.2. Future improvement prospects

We are exploring several research tracks in order to fully develop NAT's potential. Nat can easily integrate to other softwares because it is polymorphic. On that basis we are studying a few tracks to integrate the software and use it easily : plug-ins for openoffice, internet use, chaining with other tools in a complex online transcribing process (BraillePost project) or video transcription software (Advene project [1]). Developing NAT to other languages still remains to be done. Currently a community is setting up on the project and should soon give new transcribing functionalities. Integrating NAT as a transcribing service on a level close to the operating system could improve in a significant way the actual blinds' use of accessibility softwares to graphical interfaces. We believe that NAT's transcriptions will not solve all accessibility problems : we are also working on implementing interaction facilitators with trace-based reasoning and experience[5], in order that the digital document be really accessible (content, structure and possible interactions) to a blind person. These researches are undertaken in the SILEX team (http://liris.cnrs.fr/silex) and should also enlarge our action to all visually impaired.

References

[1] Aubert O., Champin P.A., Prié Y., Richard B., “Active reading and hypervideo production”, *Multimedia system journal*, Special issue on canonical processes of media production, 2008.

[2] Mackenzie, sir C.N., “World Braille usage : a survey of efforts towards uniformity of Braille notation”, *International meeting on Braille uniformity*, UNESCO, Paris, 1950, pub. 1954.

[3] Magna F., “Le braille mathématique”, *Revue bimestrielle du Groupement des professeurs et éducateurs d'aveugles et amblyopes*, n °183, 2002.

[4] Mascret B., Mille A., Ollier M., “Un transcripteur braille idéal ?”, *5ème conférence Handicap*, Paris, June 2008

[5] Mille A., From case-based reasoning to trace-based reasoning, *Annual Reviews in Control* 30, p.223-232, ELSEVIER, 2006.

[6] Moço V., Archambault D., “Automatic translator for Mathematical Braille”, *2nd International Conference on Universal Access in Human-Computer Interaction*, Heraklion, Greece, June 2003

[7] Ollier M., “Aides techniques, nouvelles technologies et scolarisation d’élèves déficients de la vue”, *La nouvelle revue de l’adaptation et de la scolarisation*, hors série n° 3, « Scolariser les élèves déficients visuels aujourd’hui », 2007.

[8] Sperandio J.-C., Uzan G., Oltra R., “L’informatique comme barrière d’exclusion ou comme aide technique à l’intégration”, *Performances Humaines et Techniques*, n° hors série : “Situation de Handicap”, p. 34-40, June 1999.

[9] Uzan G., “Temps technologiques, temps individuels, temps sociaux : l’articulation des contraintes temporelles dans l’utilisation de l’informatique par des aveugles”, *Colloque Fisaf*, Paris, 2003.

[10] Uzan G., “Ergonomie cognitive du handicap visuel : une contribution à la conception d'aides informatiques”, Thèse de doctorat, Université René Descartes Paris 5, December 9th 2005.