An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually-Impaired Users

Ali Awde[1], Manolo Dulva Hina[1,2], Chakib Tadj[1]
[1] Université du Québec, École de technologie supérieure, Montréal Québec, Canada
[2] PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France
Email: ali.awde.1@ens.etsmtl.ca, manolo-dulva.hina.1@ens.etsmtl.ca, ctadj@ele.etsmtl.ca

Yacine Bellik
Université de Paris-Sud, France LIMSI-CRNS, Université de Paris-Sud, France
Email: yacine.bellik@limsi.fr

Amar Ramdane-Cherif [2,3]
[3] LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France
Email: rca@prism.uvsq.fr

Abstract—**Presenting mathematical expressions to visually-impaired users is a challenging task because, unlike text, a mathematical expression is bi-dimensional and has some distinguishing characteristics. At present, the state-of-the art solutions present mathematical expressions to blind users in only one presentation format and with no consideration of user's context. In this work, we present mathematical expressions in more than one presentation format and consider the context of the user, his environment and this computing system as well as the nature of the expression itself and of the user's preferences. The solution presented in this work is most efficient when users are familiar with many presentation formats. Unlike the current state-of-the art solutions, ours takes into account the user's situation and present a solution that is suitable to his context and capacity. This work is our contribution to an ongoing research to make informatics more accessible to handicapped users.**

Index Terms—**visually-impaired users, mathematical presentation, multimodal multimedia computing, machine learning, human-machine interaction**

I. Introduction

Recently, various systems have been specifically developed for visually-impaired people so they can use computing systems. A number of such systems have been developed in order to allow users to access mathematical expressions. The presentation of mathematical expressions to blind users is a challenge due the following reasons: First, the visual mathematical representation, unlike that of text, is bi-dimensional and the interpretation of a mathematical expression is related to one's knowledge of the expression's spatial components. Second, the conversion of a multi-dimensional structure to a non-visual representation is not straightforward. For example, the representation of an expression, say in Braille, requires additional information to explain a specific term (for example, an exponent). Also, the conversion of a mathematical expression into a certain format, say audio, is often ambiguous. Third, the vocabulary terms used by sighted people are quite large compared to the amount of data that can be made accessible to a visually-impaired user. For example, a standard 6-dot Braille[1] can encode only 64 characters. This number of characters is not enough to represent all frequently-used mathematical symbols. Hence, a mathematical symbol itself is made represented by a combination of these characters. However, large quantity of symbols is a challenge to blind users. Finally, the manipulation of data for visually-impaired user is not as convenient as it is for sighted people. For example, Braille characters are often embossed into a paper. Once an expression is printed in a static media, the user cannot easily change the data.

Indeed, to visually-impaired users, a mere understanding of a mathematical expression requires repeated passage on the expression in which the user sometimes skips some secondary information, only to revert back to them again and again until he fully grasps the meaning of the expression. A complicated task like this is detailed in [1]. Hence, some tools were developed to lessen the complexity of performing a similar task. Among these tools are MathTalk [2], Maths [3], DotsPlus [4], EasyMath [5], and AudioMath [6, 7]. MathTalk and Maths convert a standard algebraic expression into its audio equivalent. In using Maths, the user can read, write and manipulate mathematics by using a multimedia interface that contains speech, Braille and audio. Aster (*Audio System for Technical Readings*) [8], on its part, takes in a Latex document and reads it using various

[1] http://6dotbraille.com

tones to denote different terms of the expression. VICKIE (*Visually Impaired Children Kit for Inclusive Education*) [9] and BraMaNet (***Bra**ille **Ma**thématique sur le **Net***) [10] are transcription tools that convert a mathematical document (written in Latex[2], MathML[3], etc.) to its *Braille* representation. DotsPlus is a tactile method of printing documents that incorporates both Braille representation and graphic symbols (e.g. ∏, ∑, etc.). For EasyMath, its objective is to produce a bi-dimensional output of a mathematical expression, similar to the representation for sighted people, using Braille characters and an overlay keyboard.

None of the tools cited above, however, is complete. Studies were conducted evaluating these tools based on users' needs [5, 11]. The results indicate that users are neither independent nor able to do their homework (i.e. case of students) without the help of sighted people. Indeed, each of these tools has its own set of limitations. For example, Aster uses only a LaTex document while AudioMath allows only MathML document for conversion into an audio output.

To address the weaknesses of the various existing solutions cited above, we do get the strength of each tool and integrate each one in our work to build a system that (1) extends the limitations of currently existing systems, (2) provides users with opportunities to access as many types of document as possible, and (3) presents data output in as many suitable formats as possible after considering user situation and the special symbols within the expression. This work is essential as it offers to provide all types of data presentation formats while requiring minimum explicit intervention from the user.

In this paper, we present the infrastructure of a multimodal multimedia computing system that presents mathematical expressions to visually-impaired users after taking into account the user's interaction context and preferences and the nature of the expression itself. Apart from this introductory section, the rest of this paper is structured as follows. Section II lists down the technical challenges in this work and essays our approaches to address each one of them. Section III presents the design and principle that address our system requirements. System design is provided in Section IV. In Section V, we provide examples that apply the concepts discussed in the previous section. Also, we present some formal specifications on the functionalities of our system. Finally, we present our future works and conclusion in section VI.

II. Technical Challenges

Our principal goal is to build a model of an adaptive multimodal system that correctly presents mathematical expressions to visually-impaired users. In our proposed solution, we first consider the user's interaction context as a requisite in determining the most appropriate (a.k.a. optimal) modality. Here, we define *modality* as the mode of human-computer interaction. Next, knowing the optimal input/output modalities and using the media devices' priority ranking, we determine the media devices that would support the chosen modalities. Then, we determine the nature of the mathematical expression – whether or not it contains special mathematical operation symbol (more details provided in Sections III-D and III-F). Finally, we determine the most suitable presentation format for the given mathematical expression based on the available media devices, the nature of mathematical expression and the user's preferences on presentation formats. The design of such a system needs to address the key requirements cited below:

Requirement 1: *Provide a relationship between an interaction context (i.e. combined user, environment and system context) and the suitable modalities for visually-impaired users*. What constitutes a user context given that a user is already visually-impaired? What parameters constitute the user, the environment and the system context? What are the modalities that are available to a blind user? On what basis a specific modality is selected?

Requirement 2: *Provide a relationship between modality and media devices, and a system mechanism for a seamless adaptation when a media device fails (i.e. absent or defective) or is newly added*. What relationship exists between media devices and modalities? When a media device is shut down, does it follow that the modality associated to the device must also be shut down? When there are many available media devices to support a modality, do they have to be activated all together? If not, then how are they prioritized?

Requirement 3: *Provide a mechanism for optimal selection of a mathematical expression's presentation format based on available media devices and their priority rankings relative to the modalities they support and of user's format preferences (as per presence/ absence of special operation symbol within the expression)*. What are the available presentation formats? How are they ranked? Is it possible to have two or more presentation formats activated at the same time?

Requirement 4: *Provide a learning mechanism that is applicable to the selection of modality given an interaction context. Derive a relationship that relates media devices selection with respect to the given modality. Derive a function that selects a most suitable presentation format given the available media devices, the nature of mathematical expression and the user's preferences on presentation formats*. What are the condition, the decision and the corresponding action that an intelligent multimodal system must learn? Is the learning continuous or will it stops after learning certain amount of knowledge?

We address the technical challenges by providing specific solutions to the system requirements cited above.

Proposed Solution to Requirement 1: The possible modalities left for visually-impaired users are tactile and vocal modalities (i.e. visual modality is already not suitable), both for data input and output. Throughout this paper, we will use the following modality designations:

---

[2] L. Lamport, LaTeX: The Macro Package, http://web.mit.edu/texsrc/source/info/latex2e.pdf, 1994.
3 MathML, http://www.w3.org/Math

V_{in} = *vocal input*, V_{out} = *vocal output*, T_{in} = *tactile input*, and T_{out} = *tactile output*. In a user context, further handicap (e.g. manual handicap or lack of knowledge in using Braille) affects modality selection (e.g. tactile input and output). The noise level and environment restriction also affects the possibility of using vocal input and/or output modality. "*Noise*", in acoustics and in this paper, refers to meaningless or unwanted sound that is louder than a desired volume. We classify two levels of noise – *quiet* and *noisy*. "*Environment restriction*" is a term we used to refer to whether the working place imposes mandatory silence or not. More details of the effect of noise level and environment restriction on modality selection are provided in Section IV-B. In the system context, for example, the use of PDA and cellular phone as computing device also affects the possibility of using tactile modality. The ensemble of all these context parameters (i.e. all together forms the interaction context) determines whether a modality is possible or not. Once possible, the interaction context also determines the most suitable modalities.

Proposed Solution to Requirement 2: We have a *media devices priority table* (MDPT) containing records of all media devices known to the system. Our system assigns *a priori* the type of modality that some familiar media devices support. The end users will do the same when new media devices are introduced into the system. There is a possibility, however, that two or more media devices support a common modality (e.g. both overlay keyboard and Braille terminal support tactile output mode). In such a case, the end user provides priority rankings on these media devices. The default devices (and software such as speech recognition software) are always ranked top in the priority ranking. Two or more devices, however, may share a common priority ranking, if end user so desires.

Proposed Solution to Requirement 3: The presentation format selection is resolved using a formula based on priority rankings of presentation formats and their supporting media devices. The format with the highest score is selected as the final presentation format. In case of similar scores for two (or more) presentation formats, both are selected since both are possible. The presence of special operation symbol in a mathematical expression yields a presentation format's priority ranking that is different from one that has no special symbol. Such priority ranking is obtained from user's profile.

Proposed Solution to Requirement 4: We provided a generic representation of interaction context. By using the proposed supervised learning method, our system learns what modalities are suitable to a given instance of interaction context. A relationship is established between modalities and media devices; media devices themselves are ranked by priority, the ranking being provided by the end user. This ranking is also important during the search for a replacement to a failed media device. Also, a function is derived that helps determine all possible presentation formats. By using the tables of user's preferences and the nature of the mathematical expression, we can determine the most suitable presentation format for the user. Our proposed method for knowledge acquisition is incremental, thus continuous. In this manner, the system becomes adaptive as the system evolves. This adaptation includes schemes such as an introduction of newly added media devices, a modification of interaction context parameters, etc. In general, we attempt to make our system adaptive to most possible revisions with minimum cost of modification.

III. Detailed Design and Principles for the Resolution of System Requirements

The details of the proposed solutions based on the requirements cited in the previous section are discussed in this section.

A. Using Machine Learning to Find Optimal Suitable Modalities to an Interaction Context

Machine learning (ML) is concerned with the design and development of algorithms and techniques that optimizes an entity's performance using sample data or past experience [12]. In our work, ML is adopted in order that our system learns from its previous experiences.

A *scenario* is an event that needs system's response. The set of inputs that triggers a scenario is called the *pre-condition scenario* while the state of the system after reacting to a scenario is called *the post-condition scenario*. In this work, the system acquired knowledge about scenarios is stored in a *scenario repository* (SR).

Let interaction context, $\mathbf{IC} = \{\mathbf{IC_1}, \mathbf{IC_2}, \ldots, \mathbf{IC_{max}}\}$, be a set of all possible interaction context. At any given time, a user has a specific interaction context i denoted $\mathbf{IC_i}$, $1 \leq i \leq$ max. Formally, an interaction context is a tuple composed of a specific user context (**UC**), environment context (**EC**) and system context (**SC**). An instance of **IC** may be written as:

$$\mathrm{IC_i} = \mathrm{UC_k} \otimes \mathrm{EC_l} \otimes \mathrm{SC_m} \tag{1}$$

where $1 \leq k \leq max_k$, $1 \leq l \leq max_l$, and $1 \leq m \leq max_m$, and max_k = maximum number of possible user context, max_l = maximum number of possible environment context, and max_m = maximum number of possible system context. The Cartesian product (symbol: $\otimes$) means that at any given time, **IC** yields a specific combination of **UC**, **EC** and **SC**.

The user context **UC** is made up of parameters that describe the state of the user during the conduct of his activity. Any specific user context k is given by:

$$\mathrm{UC_k} = \bigotimes_{x=1}^{max_k} \mathrm{ICParam_{kx}} \tag{2}$$

where $\mathbf{ICParam_{kv}}$ = parameter of $\mathbf{UC_k}$ where k is the number of **UC** parameters. Similarly, any environment context $\mathbf{EC_l}$ and system context $\mathbf{SC_m}$ are given as follows:

$$\mathrm{EC_l} = \bigotimes_{y=1}^{max_l} \mathrm{ICParam_{ly}} \tag{3}$$

$$\mathrm{SC_m} = \bigotimes_{z=1}^{max_m} \mathrm{ICParam_{mz}} \tag{4}$$

The first knowledge that our system's ML component must learn is to relate the interaction context to an appropriate modality. In general, a modality is possible if there exists at least one modality for data input and at least one modality for data output. Given a modality set **M = {V_{in}, T_{in}, V_{out}, T_{out}}** then modality is possible under the following condition:

$$\text{Modality Possible} = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \quad (5)$$

Hence, failure of modality can be specified by the following relationship:

$$\begin{aligned}\text{Modality Failure} = &((V_{in} = \text{Failed}) \wedge (T_{in} = \text{Failed})) \vee \\ &((V_{out} = \text{Failed}) \wedge (T_{out} = \text{Failed}))\end{aligned} \quad (6)$$

where the symbols $\wedge$ and $\vee$ denote logical AND and OR, respectively.

Let $\mathbf{M_j}$ = element of the power set of **M**, that is, $\mathbf{M_j} \in \mathcal{P}(\mathbf{M})$ where $1 \leq j \leq mod\_max$ (maximum modality). Also, let $\hat{\mathbf{M}}$ = the most suitable $\mathbf{M_j}$ for a given interaction context $\mathbf{IC_i}$. Let **X** be the set of pre-condition scenarios. Hence, the relationship between **X** and **IC** can be written as $X_i = IC_i$. For the purpose of simplicity, we let the pre-condition set **X** be represented by interaction context **IC**. Each interaction context i, IC_i, is composed of attributes of n components, that is, $IC_i = (A_1, A_2, ..., A_n)$ where attribute is a parameter that may belong to **UC** or **EC** or **SC**. We also let the set of post-condition **Y** be represented by a set of modality **M**. Let the function f map the set of **IC** to the set of **M**, in which h calculates the suitability score of such a mapping, and is given by:

$$h\left(f(IC_i) \rightarrow M_j\right) = <suitability\_score> \quad (7)$$

In Mathematics, the probability function h can also be written as:

$$h = P(M_j \mid IC_i) \quad (8)$$

which should be read as the probability of the occurrence of M_j given an IC_i. To simplify calculation, *Bayes Theorem* [13], given below, can be adopted:

$$P(M_j \mid IC_i) = \frac{P(IC_i \mid M_j) \times P(M_j)}{P(IC_i)} \quad (9)$$

The implementation of Bayes Theorem leads to the *Naive Bayes algorithm* [14]. The Naive Bayes algorithm is a classification algorithm that assumes that the IC_i attributes $A_1,\ldots,A_n$ are all conditionally independent of one another given a post condition M_j. The representation of $P(IC_i|M_j)$ becomes:

$$\begin{aligned}P(IC_i \mid M_j) &= P(A_1,...,A_n \mid M_j) \\ &= P(A_1 \mid M_j) \times ... \times P(A_n \mid M_j) \\ &= \prod_{i=1}^{n} P(A_i \mid M_j)\end{aligned} \quad (10)$$

Here, our goal is to train a classifier that, given a new IC_i to classify, will provide the probability distribution on all possible values of **M** (i.e. M_1, M_2, …, M_m). Given that $IC_i = (A_1, A_2, ..., A_n)$, then (9) becomes:

$$P(M_j \mid A_1...A_n) = \frac{P(M_j)P(A_1...A_n \mid M_j)}{\sum_{k=1}^{m} P(M_k)P(A_1...A_n \mid M_k)} \quad (11)$$

Equation (11) can also be written as:

$$P(M_j \mid A_1...A_n) = \frac{P(M_j)\prod_{i=1}^{n} P(A_i \mid M_j)}{\sum_{k=1}^{m} P(M_k)\prod_{i=1}^{n} P(A_i \mid M_k)} \quad (12)$$

which is the fundamental equation for the Naive Bayes classifier. Given a *new* instance of interaction context $IC_{new} = (A_1 \ldots A_n)$, the equation shows how to calculate the probability that M_j will take given the observed attribute values of IC_{new} and given that the distributions $P(M_j)$ and $P(A_i|M_j)$ are estimated values taken from training data (SR). If we are interested only in the *most suitable value* of M_j, then we have the Naive Bayes classification rule:

$$h_{best} = \hat{M} = \arg\max_j \left(\frac{P(M_j)\prod_{i=1}^{n} P(A_i \mid M_j)}{\sum_{k=1}^{m} P(M_k)\prod_{i=1}^{n} P(A_i \mid M_k)} \right) \quad (13)$$

Given that the denominator does not depend on parameter j, then the above equation becomes

$$h_{best} = \hat{M} = \arg\max_j \left(P(M_j)\prod_{i=1}^{n} P(A_i \mid M_j) \right) \quad (14)$$

where $P(M_j)$ = the frequency of M_j in SR ÷ cardinality of (SR).

B. Algorithms for Finding Optimal Modalities for Interaction Context

In Appendix A, Figure 12 shows the algorithm that calculates the suitability score of an element of a modality power set based on a given interaction context. Figure 13, on the other hand, finds the optimal modality for the given instance of interaction context.

Given that **M = {V_{in}, T_{in}, V_{out}, T_{out}}**, then the power set (i.e. the set of all subsets) of **M** is given by $\mathcal{P}(\mathbf{M})$ = {**{V_{in}}, {T_{in}}, {V_{out}}, {T_{out}}, {V_{in}, T_{in}}, {V_{in}, V_{out}}, {V_{in}, T_{out}}, {V_{in}, T_{in}, V_{out}}, {V_{in}, T_{in}, T_{out}}, {V_{in}, V_{out}, T_{out}}, {V_{in}, T_{in}, V_{out}, T_{out}}, {T_{in}, V_{out}}, {T_{in}, T_{out}}, {T_{in}, V_{out}, T_{out}}, {V_{out}, T_{out}}, {}**}. $\mathbf{M_j}$, therefore, evaluates the suitability score of each element of $\mathcal{P}(\mathbf{M})$, see Figure 12. The optimal modality, $\hat{M} = h_{best}$, is then chosen from one of these elements (see Figure 13). The selected element is one that satisfies (14) and one having the highest suitability score.

C. Relationship between Modality and Media Devices

Once the optimal (input/output) modalities (i.e. $\hat{M}$) are chosen for a given interaction context, the next step is to determine if media devices are available to support the chosen modalities. If so, then these devices are activated. Otherwise, the chosen optimal modalities are reevaluated, that is, the modality that cannot be supported by media devices is taken out from the selection. Afterwards, the result is again evaluated to determine if (14) still holds. If the answer is affirmative, then modality is possible and the result is implemented. Otherwise, there is a failure of modality. Figure 14 (see Appendix A) shows the algorithm that implements the update of optimal modalities based on availability of media devices.

There are usually more than one media devices that support a specific modality. For example, a regular keyboard, an overlay keyboard and a Braille terminal all support tactile input modality. Activating them all is just plain redundancy; hence the system must select one from the given available devices. If these devices have equal chances of being selected, then device selection becomes random, and the chance that device Di gets chosen out of n available devices is given by:

$$P(D_i / M_j) = 1/n \quad (15)$$

In real world, however, the user has some preferences for media devices. And such user's preferences can vary from one user to another. Hence, we invoke priority ranking such that when many media devices supporting a chosen modality are available, only the top-ranked media device is activated. Also, priority ranking is essential in finding replacement to a failed media device. Usually, when a media device is malfunctioning or absent (a.k.a. *failed*), the system searches the device (that supports the same modality) which is next in priority. If it is found, the replacement device is activated and the search is over. Otherwise, the system keeps searching for a replacement through *priority ranking order*.

Given a media device D_i, where $1 \le i \le n$ and i = priority index and n = number of media devices supporting modality M_j, then the probability that D_i being adopted as the media device that will support modality M_j is given by:

$$P(D_i / M_j) = 1 - \sum_{1}^{i-1} (1/n) \quad (16)$$

Let there be a *media devices priority table* (MDPT) (see Table I) containing media devices grouped according to the modality they support and arranged by priority ranking. When our system implements a modality, it selects the media device(s) that is/are ranked top in priority. It is also through the MDPT that the system searches for a replacement to a failed media device.

TABLE I.
A SAMPLE MEDIA DEVICES PRIORITY TABLE (MDPT)

Modality	Media Devices by Priority			
	1	2	3	n
Vocal input (V_{in})	microphone, speech recognition			
Vocal output (V_{out})	speaker, speech synthesis	headset		
Tactile input (T_{in})	keyboard	overlay; Braille terminal	Vmouse	
Tactile output (T_{out})	Braille terminal	tactile printer		

When a *new media device* d_{new} is *added* or *introduced* to the system for the *first time*, the device is associated to a modality and is given a priority ranking **r** by the user. What happen to the rankings of other devices d_i, $(1 \le i \le n$, and n = number of media devices) which are in the same modality as d_{new} in the MDPT? Two things may happen, depending on the user's selection. The first possibility is that after having the new device's priority **Priority(d_{new})** set to ***r*** then the priority of the other device **i,** $(1 \le i \le n)$ denoted **Priority(d_i),** remains the same. The second possibility is the priority rankings of all media devices ranked **r** or lower are adjusted such that their new priority rankings are one lower than their previous rankings. Formally, in Z [15], this is specified as: $\forall$**i,** $\exists$**r: ℕ;** $\forall$**d_i,** $\exists$**d_{new}: Devices | (Priority(d_{new}) = r ∧ Priority(d_i) ≥ r) ⇒ Priority(d_i)' = Priority(d_i) + 1.**

Given the familiar media devices, the association between modalities and media devices can be denoted as:

$$V_{in} = (\text{Microphone} \wedge \text{Speech recognition}) \quad (17)$$

$$V_{out} = \text{Speech synthesis} \wedge (\text{Speaker} \vee \text{Headset}) \quad (18)$$

$$T_{in} = (\text{Keyboard} \vee \text{Braille Terminal} \vee \text{Overlay Keyboard} \vee \text{Vmouse}) \quad (19)$$

$$T_{out} = (\text{Braille Terminal} \vee \text{Tactile Printer}) \quad (20)$$

D. Mathematical Expression and Presentation Formats

To a visually-impaired user, a *simple* mathematical expression becomes complex due to the presence of elements such as the *subscript*, *exponent*, *mathematical symbols* (e.g. ∏, ∑, etc.), and the expression's *dimension* (i.e. complex numerator, denominator). In informatics, a mathematical expression is generally represented in *MathML* format which is an application of XML for describing mathematical notations and capturing its structure and content and aimed at integrating them into World Wide Web documents. For example, in Figure 1, a simple fraction is shown with its equivalence in MathML, in Braille and its linear representations.

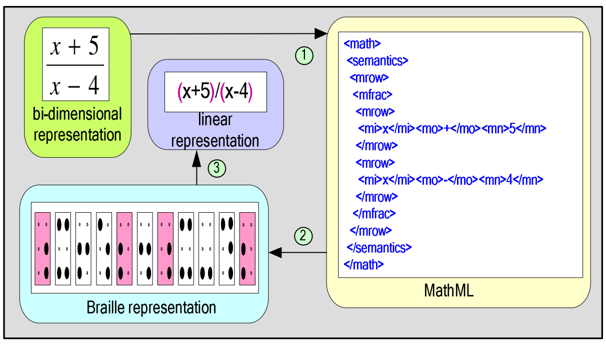


Figure 1: A fraction converted into its linear representation, and its corresponding MathML and Braille representations.

Different presentation formats use different methods to represent a mathematical operation. For example, in using

Braille, there is a unique symbol for every operation. In using *speech*, an operation is uttered using a specific word (e.g. "+" is "plus", "-" is "minus", etc). Using *DotsPlus*, an operation is represented by a unique symbol in tactile form. Using *EasyMath*, every basic operation (e.g. +, -, ×, ÷, etc.) is represented by a unique symbol similar to its Braille representation. For *special operation* (e.g. ∑, ∏, log, ∫, ∬, ∭, etc), however, its representation is in tactile form. Note that the representation of special operations in DotsPlus and EasyMath are not the same. For example, the + operation symbol is represented by a Braille symbol (see Figure 1) whereas in DotsPlus it is represented as "+" in embossed tactile form.

E. Analysis of a Mathematical Expression

A mathematical expression is analyzed as follows. Given a mathematical expression in MathML format, it is analyzed lexically using *grammar rules and dictionary*. The result yields a list of lexemes. A lexeme is a parameter within an expression which may be an operand or an operator. Given the lexemes, the *parser* analyzes the expression parameters (i.e. operands and operations) then sends the operation(s) to the *Expression Evaluator* to determine if special symbol is present while at the same time all the parameters are sent to the *Expression Encoder* to translate the expression into its encoded format. This process is shown in Figure 2.

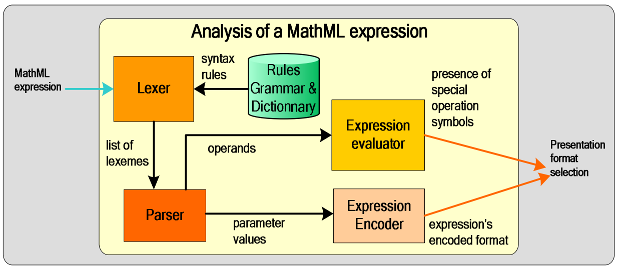


Figure 2: Structure of the analysis of a MathML expression.

As an example, the analysis of our specimen expression, the fraction defined in Figure 1, is shown in Figure 3. As shown, in step 1, the MathML expression is sent to the Lexer. In step 2, using the XML grammar, the expression is decomposed into a list of lexemes; the list is then sent to the parser. In step 3, the operations and operands in the expression are sent to expression evaluator and encoder. Together, in step 4, the evaluator deduces the presence of special symbol (e.g. fraction bar) while the encoder produces the encoded expression. Finally, in step 5, the presence of special operation symbol is detected and the encoded expression becomes an input to the presentation format selection process.

F. Selection of Presentation Format

In this work, we consider 4 presentation formats which are currently in existence, namely: the Braille, speech, DotsPlus and EasyMath.

The *Braille* format presents a mathematical expression in linear Braille representation. This format "linearizes" mathematics. Note that any bi-dimensional mathematical expression can be encoded as linear expression by using parentheses. A user must be familiar with Braille in order to use this format. Some projects, such as VICKIE and BraMaNet, built translators that convert expression (in Latex or MathML format) to its Braille equivalent.

Speech, on the other hand, is the modality that is used daily by visually-impaired people. For mathematical application, speech converts an expression into its audio equivalent. Some tools that use speech as presentation format are MathTalk, Aster and AudioMath.

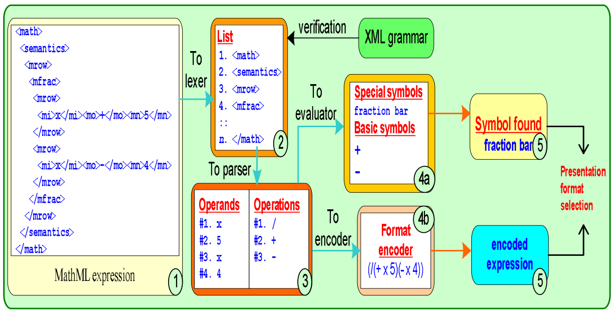


Figure 3: A sample analysis of a specimen fraction.

DotsPlus is an application that allows the printing of documents that contains mathematical symbols (e.g. ∏, ∑, etc.). These symbols are encoded as graphics shaped exactly as how they looked. When a blind user touches the embossed symbol, the user senses the bi-dimensionality of an expression. If the user intends to use DotsPlus, he must also be familiar with Braille since this specific format also uses Braille characters.

The objective of *EasyMath* is to produce a bi-dimensional output of a mathematical expression, similar to the representation for sighted people, using Braille characters and an overlay keyboard. EasyMath incorporates both audio and tactile forms. The use of EasyMath requires that the user uses an overlay keyboard and must be familiar with Braille.

The *media devices* are also factors in determining what presentation format should be used to present a mathematical expression. The four presentation formats, therefore, as functions of media devices, are given by:

$$\text{Speech presentation format} = (\text{Speech Synthesis}) \wedge (\text{Speaker} \vee \text{Headset}) \quad (21)$$

$$\text{Braille presentation format} = (\text{Braille Terminal} \vee \text{Tactile Printer}) \quad (22)$$

$$\text{EasyMath presentation format} = (\text{Overlay Keyboard} \wedge \text{Tactile Printer}) \wedge (\text{Speaker} \vee \text{Headset}) \quad (23)$$

$$\text{DotsPlus presentation format} = \text{Tactile Printer} \quad (24)$$

As shown in Figure 4, the selection of the *suitable presentation format* begins with the given interaction context (step 1). The ML component, with reference to the modality SR (step 2), determines the optimal modalities (step 3). The selected optimal modalities as well as the available media devices (step 4) serve as input to the media selection process. With reference to the equations shown in Section III-C, the selected media devices are determined (step 5). These devices along with the encoded expression (step 6-a), the presence/absence of special mathematical operation symbol (step 6-b), and the presentation formats available in the user's computing system (step 6-c) are all inputs to the presentation format selection process. The output is the selected optimal presentation format. With reference to the specimen fraction expression in Figure 1, the selected presentation format is Braille. The diagram also illustrates the media devices that support Braille presentation format and the equation's corresponding translated expression in various formats.

In the same figure, we also demonstrate the various *media devices that support other presentation formats* (namely, speech, DotsPlus and EasyMath).

Given selected modalities and available media devices, it is possible that two or more presentation format(s) may be suitable to a given MathML mathematical expression. How do we determine then the optimal presentation format? To resolve this issue, we adopt that the selection of optimal presentation format shall be based on user's preferences, on priority ranking that he/she assigned to different presentation formats and to media devices.

The MDPT is used in determining the priority of media devices as per modality that they support. Table II illustrates a sample of a *presentation format priority table* (PFPT). As the table shows, there is a separate priority ranking for presentation format when the mathematical expression contains a special operation symbol. It also has another priority ranking for presentation format if the expression contains no special operation symbol. Examples of basic operation symbols include +, -, ×, ÷, etc. Examples of *special operation symbols* include $\sum$, $\prod$, log, $\int$, $\iint$, $\iiint$, etc.

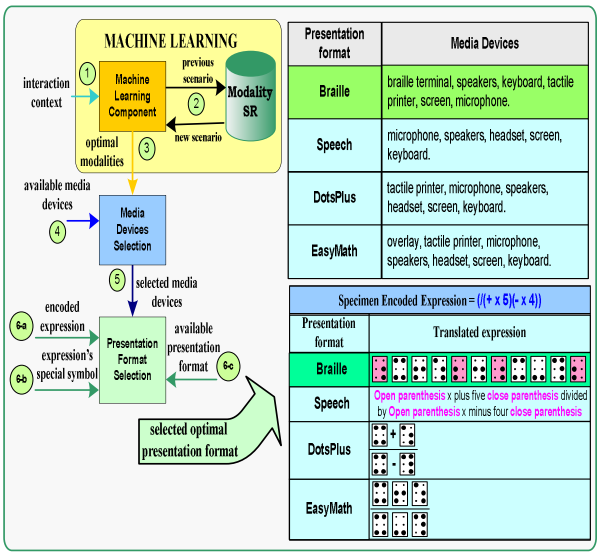


Figure 4: The presentation format selection process based on the given interaction context and MathML expression.

The optimal presentation format selection process uses the following functions to determine the score of each presentation format. The one that obtains the highest score is considered as the optimal presentation format:

$$\text{Presentation Format Score} = \text{Presentation Format Priority} \times \text{Media Devices Priority} \quad (25)$$

$$\text{Presentation Format Priority} = (m + 1 - p) / m \quad (26)$$

$$\text{Media Devices Priority} = \frac{\sum_{i=1}^{n} (n_i + 1 - d_i) / n_i}{n} \quad (27)$$

such that for presentation format priority, the variable m = number of suitable formats and p = priority ranking as obtained from PFPT, $1 \le p \le m$. For priority of media devices, n = available media devices supporting a presentation format (see (21) through (24)). Also given i^{th} device, where $1 \le i \le n$, then d_i = priority ranking obtained from MDPT and n_i = number of media devices supporting the same modality as the i^{th} device. Figure 5 shows the algorithm that implements the selection of suitable presentation format.

```
Suitable Presentation Format
Input:
1. All available media devices Di (1 ≤ i ≤ n)
2. Presentation format Fj (1 ≤ j ≤ m)
3. Media devices priority table (MDPT)
4. Presentation format priority table (PFPT)
Output: Suitability of Presentation format Fj
Procedure:
1. for each Presentation format Fj do
2.   get media devices ML supporting Fj(1 ≤ L ≤ nL)
3.   get priority p of Fj (from PFPT)
4.   get priority pL of media ML (from MDPT)
5.   PFP = (m + 1 - p) / m (i.e. equation 26)
6.   sum = 0
7.   for k = 1 to n do
8.        sum = sum +((nk + 1 - dk)/ nk) (i.e. equation 27)
9.   endfor
10.  MDP = sum / n
11.  PFS = PFP * MDP
12. endfor
13. return PFS
Endprocedure
```

Figure 5: Algorithm for finding the suitable presentation format for a mathematical expression.

TABLE II.
PRESENTATION FORMAT PRIORITY TABLE

Presentation Format by Priority	
Expression with special symbol	Expression without special symbol
1. EasyMath 2. DotsPlus 3. Braille 4. Speech	1. Braille 2. Speech 3. EasyMath 4. DotsPlus

IV. DETAILED DESIGN AND PRINCIPLES FOR THE RESOLUTION OF SYSTEM REQUIREMENTS

The details of the infrastructure satisfying the design specifications cited earlier are explained in this section.

A. Architectural Framework

Figure 6 shows the layered view of our adaptive multimodal computing system for visually-impaired users. Adopting a layered architecture approach, one in which data moves from one defined level of processing to another, helps confine any error made during debugging to a specific layer and prevents the ripple effect of error propagation to other system components. The functionalities of each layer are given below:

- *Context Gathering Layer* – detects current interaction context;
- *Control and Monitoring Layer* – controls the system, coordinates the detection of interaction context, the mathematical expression, its presentation and/or manipulation;
- *Data Analysis Layer* – here, the presentation format of the mathematical expression is selected based on available resources and user's context;
- *Data Access Layer* – allows search/edit of mathematical expression; and
- *Presentation Layer* – presents mathematical expression via optimal presentation format.

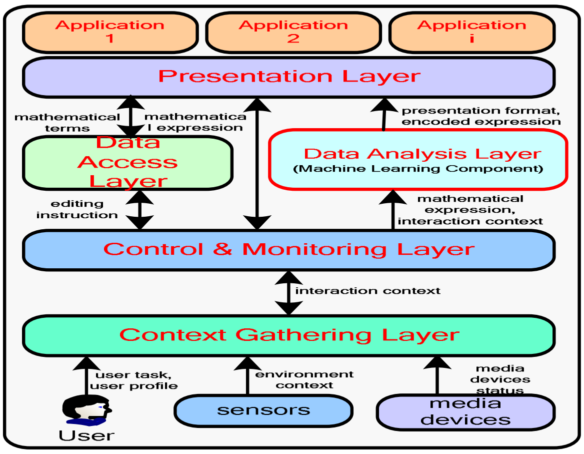


Figure 6: Architectural layer view of our multimodal computing system for visually-impaired users.

An *agent* is a preferred programming technique because the traditional techniques (i.e. functional or object-oriented programming) are inadequate in developing tools that react to environment events. Recently, multi-agent systems (MAS) [16, 17] have been widely used, from relatively small systems such as email filters up to large, open, complex, mission-critical systems such as air traffic control [18]. Some works on MAS for visually-impaired users include [19, 20]. In contrast, in our work, our main goal is the correct representation of a mathematical expression while providing users autonomy. A multi-agent system is used to implement the architectural framework of Figure 6. The functionalities of various agents in our multimodal system are shown in a tabular format in Table III.

B. *Interaction Context Specification*

Interaction context is formed by combining the context of the *user*, his *environment*, and his computing *system*. The *user context*, in this work, is a function of user profile (including any handicap) and preferences. A sample user profile, in generic format, is shown in Figure 7. The user's special needs determine other affected modalities (i.e. the user is already disqualified from using visual input/output modalities). For example, being mute prevents the user from using vocal input modality.

TABLE III.
FUNCTIONALITIES OF VARIOUS AGENTS IN OUR ADAPTIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

Layer	Agent	Functionality
Context Gathering Layer	User Agent	Detects user preferences, such as media and presentation format
	Environment Agent	Detects the user workplace's noise level and its noise level restriction
	Device Manager Agent	Detects available and functional media devices
Control and Monitoring Layer	System Management Agent	Detects user's interaction context and mathematical expression, and determines how it will be treated
Data Analysis Layer	Analysis and Conversion Agent	Obtains MathML expression and converts it to its encoded format
	Machine Learning Agent	Determines the optimal présentation format for the given user situation
Data Access Layer	Searching and Editing Agent	Obtains editing instructions from the user
Presentation Layer	Translation Agent	Converts the encoded format of an expression into its final presentation format.

Identity	Special Needs		Computing Device
username: <username>	Manually-Disabled :	<Yes/No>	computing device1 : <MAC address 1>
	Mute :	<Yes/No>	::
password: <password>	Deaf :	<Yes/No>	computing device n : <MAC address n>
	Unfamiliarity with Braille:	<Yes/No>	

Figure 7: A sample user profile.

As a function of modality, the user context **UC** can be represented by a single parameter, that of the user's *special needs*. This parameter is a *4-tuple* representing additional handicaps, namely the *manual disability*, *muteness*, *deafness*, and *unfamiliarity with Braille*. Each handicap affects user's suitability to adopt certain modalities. The convention for this special need is shown in Table IV. Hence, the failure of modality, adopted from (6), with respect to **UC** parameters is:

$$\begin{aligned}&\text{Modality Failure} = ((\text{Manually-Disabled}) \wedge (\text{Mute}))\\&\vee ((\text{Manually-Disabled} \vee \text{Unfamiliar with Braille})\\&\wedge (\text{Deaf}))\end{aligned} \quad (28)$$

TABLE IV.
THE CONVENTION TABLE OF USER'S ADDITIONAL HANDICAP AND ITS EFFECT ON USER'S MODALITY DISPOSITION

Convention No.	Manually Disabled	Mute	Deaf	Unfamiliarity with Braille	Inappropriate Modality
1	0	0	0	0	-
2	0	0	0	1	T_{out}
3	0	0	1	0	V_{out}
4	0	0	1	1	T_{out}, V_{out}
5	0	1	0	0	V_{in}
6	0	1	0	1	V_{in}, T_{out}
7	0	1	1	0	V_{in}, V_{out}
8	0	1	1	1	T_{out}, V_{in}, V_{out}
9	1	0	0	0	T_{out}, T_{in}
10	1	0	0	1	T_{out}, T_{in}
11	1	0	1	0	T_{in}, T_{out}, V_{out}
12	1	0	1	1	T_{in}, T_{out}, V_{out}
13	1	1	0	0	T_{in}, T_{out}, V_{in}
14	1	1	0	1	T_{in}, T_{out}, V_{in}
15	1	1	1	0	T_{in}, T_{out}, V_{out}, V_{out}
16	1	1	1	1	T_{in}, T_{out}, V_{out}, V_{out}

The *environment context* **EC** is the assessment of a user's workplace condition. To a blind user, a parameter such as light's brightness has no significance, while others, such as noise level, are significant.

In this work, the environment context is based on the following parameters: (1) the workplace's *noise level* – identifies if it is quiet/acceptable or noisy, and (2) the *environment restriction* – identifies whether a workplace imposes mandatory silence or not. Based on the specified parameters, the environment context, therefore, is formally given by the relationship:

$$EC = (\text{Noise Level}) \wedge (\text{Environment Restriction}) \quad (29)$$

Table V shows the affected modality based on environment's context. It also shows the convention table we have adopted for **EC**.

TABLE V.
THE CONVENTION TABLE OF AFFECTED MODALITIES BY COMBINED NOISE LEVEL AND ENVIRONMENT RESTRICTION

Environment Context		Inappropriate Modality
Value	Convention	
1	(Environment Restriction = Silence Optional) ∧ (Noise Level = Noisy)	Vocal input
2	(Environment Restriction = Silence Optional) ∧ (Noise Level = Acceptable)	
3	(Environment Restriction = Silence Required) ∧ (Noise Level = Acceptable)	Vocal input, Vocal output (speaker)
4	(Environment Restriction = Silence Required) ∧ (Noise Level = Noisy)	Vocal input, Vocal output (speaker)

The unit of noise is *decibel* (dB). In our work, *50 dB or less* is considered "*acceptable*" or "*quiet*" while *51 dB or more* is considered "*noisy*". In our system, this range can be modified, using user interface, by the end user based on his perception. The noise level is the result of an interpretation from sampled raw data taken from a sensor. An example of a noise-detection sensor is *PASPORT PS2100* [4]. In general, when the user's workplace is noisy, the effectiveness of vocal input modality is doubtful; hence an alternative modality is necessary.

In an environment where silence is required, sound-producing media (e.g. speaker) needs to be muted or deactivated. For an environment's noise restriction, we have defined a database of pre-defined places (e.g. library, park) and their associated noise restrictions (e.g. library: silence required; park: silence optional). User can modify some database records. Also, new ones can be added through the user interface.

In our work, the *system context* (**SC**) implies the user's computing device and the available media devices. The system context is managed by the *Device Manager Agent* (DMA). See Figure 8. The computing device (e.g. PC, laptop, PDA, cellular phone) also affects the modality selection. For example, using a PDA or cell phone prevents the user from using tactile input or output modality. Some of the most commonly-used media devices suitable for blind users are: (i) *Keyboard*; (ii) *Microphone*; (iii) *Speech Recognition*; (iv) *Speech Synthesis*; (v) *Speaker*; (vi) *Headset*; (vii) *Braille Terminal*; (viii) *Overlay or Concept Keyboard*; and (ix) *Tactile Printer or Embosser*.

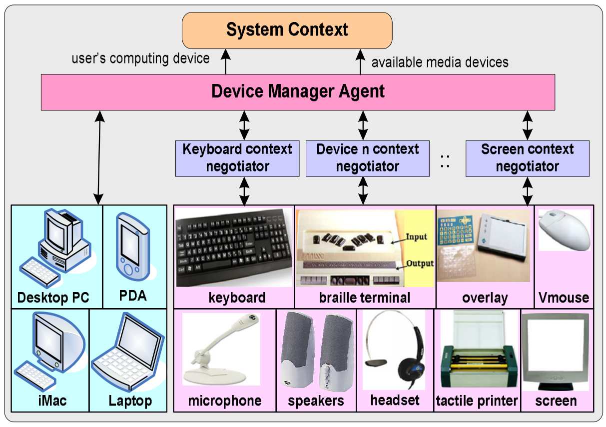


Figure 8: The DMA, responsible for detecting user's computing device and available media devices.

Let **SC**, for the purpose of modality selection, be represented by a single parameter, the user's computing device. Let **Computing Device = {PC, MAC, Laptop, PDA, Cellular phone}**. The computing device convention is shown in Table VI. For example, **SC** = 1 means that the user's computer is either a PC, a laptop or a MAC. Note that when **SC** = 2 (i.e. PDA), $\mathbf{T_{in}}$ **= Failed** because the computing device has no tactile input device; its $\mathbf{T_{out}}$ **= Failed** because, in a regular set-up, it is not possible to attach a tactile device (e.g. Braille terminal) onto it.

TABLE VI.
THE CONVENTION TABLE OF USER'S COMPUTING DEVICE AND ITS EFFECT ON MODALITY SELECTION

Convention No.	Computing device	Inappropriate Modality
1	PC/MAC/Laptop	-
2	PDA	T_{in}, T_{out}
3	Cellular Phone	T_{out}

Using ML, the following are the derived formulas on modality failures given the parameters of interaction context:

$$V_{in}\ \text{Failure} = (\text{User} = \text{Mute}) \vee (\text{Noise Level} = \text{Noisy}) \vee (\text{Environment Restriction} = \text{Silence required}) \quad (30)$$

$$V_{out}\ \text{Failure} = (\text{User} = \text{Deaf}) \quad (31)$$

$$T_{in}\ \text{Failure} = (\text{User} = \text{Manually-Disabled}) \vee (\text{ComputingDevice} = \text{PDA}) \quad (32)$$

$$T_{out}\ \text{Failure} = (\text{User} = \text{Manually-Disabled}) \vee (\text{User} = \text{Unfamiliar with Braille}) \vee (\text{Computing Device} = \text{PDA}) \vee (\text{Computing Device} = \text{Cellphone}) \quad (33)$$

4 PASPORT PS2100 Noise Detector http://store.pasco.com/pascostore/

IV. FORMAL SPECIFICATION, SIMULATION AND EXAMPLES

Here, we cited examples to illustrate how equations in previous sections are used. Formal specifications using Petri Net are also demonstrated.

A. *Formal Specifcation and Petri Net*

A *formal specification* is a mathematical description of a system that may be used to develop its implementation. Through specification, formal verification techniques can be used to demonstrate that the system design is correct with respect to the given specification.

Petri Net [5] is a formal, graphical, executable technique for the specification and analysis of a concurrent, discrete-event dynamic system. Petri nets are used in deterministic and in probabilistic variants; they are good means to model concurrent or collaborating systems. They also allow for different qualitative or quantitative analysis that can be useful in safety validation.

A Petri-Net diagram is represented by an ellipse called *place* (i.e. a state), a rectangle called *transition* (i.e. a process) and an arc representing *input* for a transition to take place (either from a place to a transition, or from a transition to a place). Places can contain *tokens*; the current state of the modeled system (the *marking*) is given by the number of tokens (and type, if they are distinguishable) in each place. Transitions are active components. They model activities which can occur (i.e. the transition fires). When the transition fires, it removes tokens from its input places and adds some to all its output places. The number of tokens removed/added depends on the cardinality of each arc.

A Petri Net specification can be defined as a quadruple (***P, T, F, B***), where **P** = a non-empty set of places, **T** = a non-empty set of transitions, ***F:*** $P \times T \rightarrow \mathbb{N}$ is the forward incidence function, ***B:*** $P \times T \rightarrow \mathbb{N}$ is the backward incidence function and $\mathbb{N}$ = the set of integers ≥ 0. **F** and **B** can be represented by forward and backward incidence matrices C(**F**) and C(**B**). A marking of a Petri Net is a mapping ***M:*** $P \rightarrow \mathbb{N}$. The "*firing*" of any transition changes the marking of the Petri Net.

In specification using Petri Net, we capture only the simulations' final results. In general, however, when a dynamic event is simulated, all possible outcomes can be obtained but due to space constraints, only a snapshot of one of many outcomes is shown. We use HPSim[6] to create, edit and simulate Petri Nets. The Petri Net diagram illustrates all possible cases that could arise, given the places and transitions of a scenario. A snapshot of one of these possible cases is shown in each of the following simulations. The diagrams are mechanisms to validate the results obtained from the simulation.

B. *Sample Case 1*

Consider, for example, an interaction context that is composed of the following parameters: **IC** = (A_1, A_2, A_3, A_4, A_5, A_6, A_7) wherein

A_1 = {true | false} = if user is manually disabled,

A_2 = {true | false} = if user is mute,

A_3 = {true | false} = if user is deaf,

A_4 = {true | false} = if user is familiar with Braille,

A_5 = {quiet | noisy} = environment's noise level,

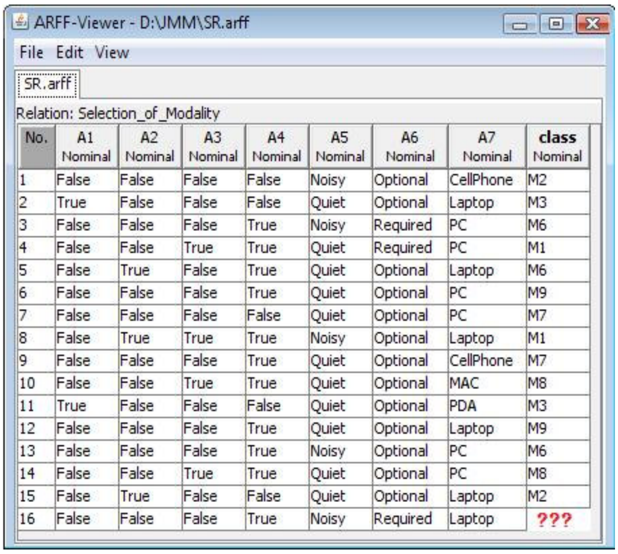
A_6 ={silence required | silence optional}
= environment's noise level restriction, and

A_7 = {PC or Laptop or MAC | PDA | Cell phone}
= user's computing device.

The set of possible modalities (i.e. refer to (5)) is given by M = {M_1,M_2,M_3,M_4,M_5,M_6,M_7,M_8,M_9} wherein $\mathbf{M_1} = \{T_{in}, T_{out}\}$; $\mathbf{M_2} = \{T_{in}, V_{out}\}$; $\mathbf{M_3} = \{V_{in}, V_{out}\}$; $\mathbf{M_4} = \{V_{in}, T_{out}\}$; $\mathbf{M_5} = \{V_{in}, T_{out}, V_{out}\}$; $\mathbf{M_6} = \{T_{in}, T_{out}, V_{out}\}$; $\mathbf{M_7} = \{T_{in}, V_{in}, V_{out}\}$; $\mathbf{M_8} = \{T_{in}, V_{in}, T_{out}\}$; $\mathbf{M_9} = \{T_{in}, V_{in}, T_{out}, V_{out}\}$. In this example, let us assume the following interaction context: (i) user context: blind with no further handicaps, familiar with Braille; hence $\mathbf{A_1}$ = False, $\mathbf{A_2}$ = False, $\mathbf{A_3}$ = False, $\mathbf{A_4}$ = True (ii) environment context: the user is in a classroom, then $\mathbf{A_5}$ = noisy, $\mathbf{A_6}$ = silence required (iii) system context: the user works on a laptop; $\mathbf{A_7}$ = Laptop. Also, the user wishes to read the following expression:

$$\int_2^3 \frac{x+1}{x-1} dx$$

The system now finds the modality that suits the given interaction context. The system does so using the principles discussed in Section III-B. Let us assume that a certain multimodal computing system's SR contains recorded scenarios as shown in Figure 9. The given figure is generated by using WEKA (*Waikato Environment for Knowledge Analysis*) [21] which is a collection of machine learning algorithms for data mining tasks. It is used in testing a machine learning algorithm as it contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.



No.	A1 Nominal	A2 Nominal	A3 Nominal	A4 Nominal	A5 Nominal	A6 Nominal	A7 Nominal	class Nominal
1	False	False	False	False	Noisy	Optional	CellPhone	M2
2	True	False	False	False	Quiet	Optional	Laptop	M3
3	False	False	False	True	Noisy	Required	PC	M6
4	False	False	True	True	Quiet	Required	PC	M1
5	False	True	False	True	Quiet	Optional	Laptop	M6
6	False	False	False	True	Quiet	Optional	PC	M9
7	False	False	False	False	Quiet	Optional	PC	M7
8	False	True	True	True	Noisy	Optional	Laptop	M1
9	False	False	False	True	Quiet	Optional	CellPhone	M7
10	False	False	True	True	Quiet	Optional	MAC	M8
11	True	False	False	False	Quiet	Optional	PDA	M3
12	False	False	False	True	Quiet	Optional	Laptop	M9
13	False	False	False	True	Noisy	Optional	PC	M6
14	False	False	True	True	Quiet	Optional	PC	M8
15	False	True	False	False	Quiet	Optional	Laptop	M2
16	False	False	False	True	Noisy	Required	Laptop	???

Figure 9: A sample snapshot of a scenario repository (SR).

As shown in the diagram, there are already 15 scenarios representing the system's acquired knowledge. The 16^{th} scenario represents a new case. Using (14), and with reference to the given interaction context and SR, the suitability score of $\mathbf{M_j}$ (where j = 1 to 9) can be calculated.

5 http://www.winpesim.de/petrinet

6 "HPSim. http://www.winpesim.de

Let us consider, for instance, the calculations involved with modality M_1:

Suitability_Score (M1) = $P(A_1 = False \mid M_1) \times P(A_2 = False \mid M_1) \times \ldots \times P(A_7 = Laptop \mid M_1) \times P(M_1) = 1 \times 0.5 \times 0 \times \ldots \times 2/15 = 0$

wherein $P(A_1 = False \mid M_1) = 2/2$, $P(A_2 = False \mid M_1) = ½$, $P(A_3 = False \mid M_1) = 0/2$, $P(A_4 = True \mid M_1) = 2/2$, $P(A_5 = Noisy \mid M_1) = ½$, $P(A_6 = silence\ required \mid M_1) = ½$, and $P(A_7 = Laptop \mid M_1) = ½$. Also, $P(M_1) = 2/15$.

Similarly, we do calculate the suitability score of all other remaining modalities. Using the same procedure, the modality that yields the highest suitability score is **M_6**:

Suitability_Score (M6) = P(A1 = False | M6) × P(A2 = False | M6) × … × P(A7 = Laptop | M6) × P(M6) = 1 × 2/3 × 1 × 1 × 2/3 × 1/3 × 1/3 × 3/15 = 0.00976.

By applying the ML algorithm (see Figure 13), **M_6** appears to respect the conditions imposed in (5), hence, it is chosen as the optimal modality for the given **IC**. This new scenario will then be added to SR as a newly-acquired knowledge (i.e. as scenario #16 in SR).

C. Simulation 1

As shown in Figure 10, the combination of interaction context's parameters yields the implementation of some modalities (i.e. M_1, …, M_9). The Net in the diagram illustrates the snapshot simulation of the case cited in sub-section V-B. As shown, the simulation begins with a token in "*Modality*" place and "*Interaction Context*" place. The firing of the token in Interaction Context yields a specific value for "*User Context*", "*Environment Context*" and "*System Context based on Computing Device*" places, which is exactly similar to the values of A_1,…, A_7. The traversal of tokens in different places is noted by green coloured places. As shown, the result yields modality M_6 being selected as the optimal modality. The Petri Net simulation confirms the result obtained in the previous section. Also, the same case yields a **V_{in}** failure result (i.e. due to noisy environment).

D. Sample Case 2

Now, consider further that the currently available media devices and their corresponding priority rankings are as shown in Table I. Also, assume that the priority rankings of user's preferred presentation formats are as shown in Table II. Based on (21) through (24), the results indicate that all four presentation formats are possible. To determine the optimal presentation format, (25) and Tables I and II are used to calculate the scores of each format. The calculation results are as follows: (i) *EasyMath* = (4/4) × (1 + (2/3) + 1) / 3 = 8/9, (ii) *Braille* = (¾) × 1 = ¾, (iii) *DotsPlus* = (2/4) × 1 = ½ and (iv) *Speech* = (¼) × 1 = ¼. Hence, the optimal selection is *EasyMath*.

E. Simulation 2

After determining the optimal modality (i.e. M_6) in the previous simulation, the system next determines the availability of the presentation formats and the media devices that will support the selected optimal modality. In the Petri Net simulation diagram of Figure 11, the status of some media devices (e.g. overlay keyboard, Braille terminal, etc.) are simulated (i.e. either OK or failed). The presentation formats are either existing or not, but for consistency purposes with the cited case, the presentation formats are simulated as all available. The diagram indicates under which conditions a specific presentation format may be availed. In the simulated diagram, the Braille, speech, DotsPlus and EasyMath presentation formats are possible. As stated, the final selection depends on user's preferences and the available media devices. In this simulated case, EasyMath obtained the highest score hence it is selected as the optimal presentation format for the given mathematical expression (see marked place).

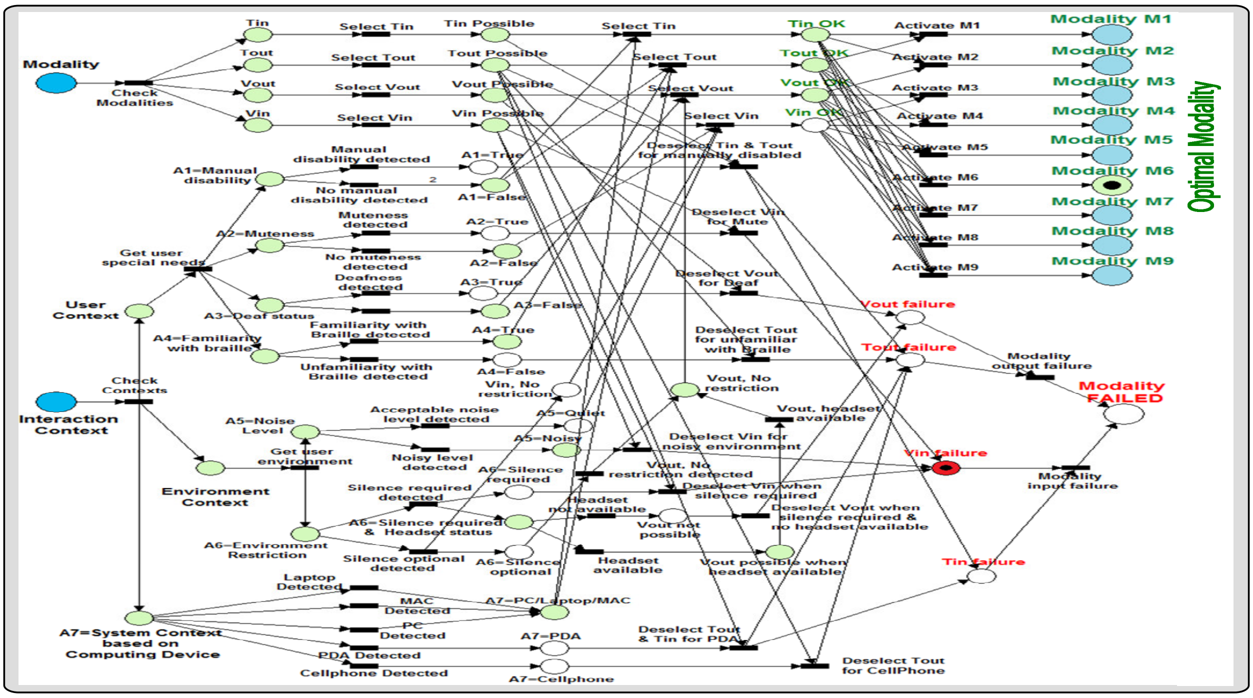


Figure 10: A snapshot of the simulated interaction context case in Section V.

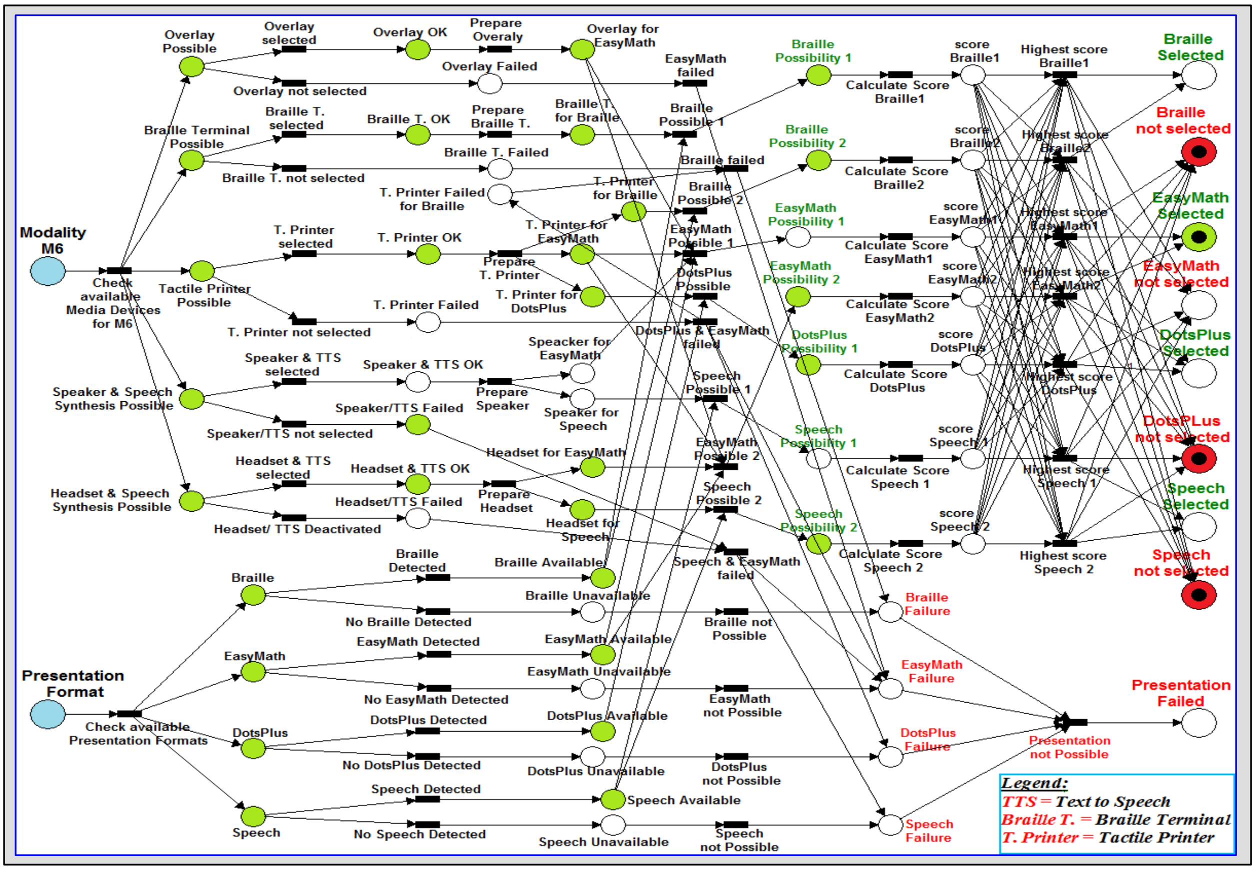


Figure 11: Selection of the optimal presentation format based on available media devices and presentation formats.

IV. Conclusion

Our ongoing research focuses on providing computing infrastructure to visually-impaired users through multimodality. One area of such domain is the infrastructure for mathematical presentation to blind users which this paper addresses. In this paper, we have noted the weaknesses of the current state-of-the art solutions. We note that the currently available solutions present mathematical expressions in one presentation format alone with no consideration on the context of the user, his working environment and his computing system as well as the nature of the mathematical expression itself and of the user's preferences. Moreover, these solutions do not provide visually-impaired users with autonomy as the users still need the assistance of sighted people.

To address the weaknesses cited above, we propose an infrastructure of a multimodal system that allows the presentation of mathematical expressions in as many presentation formats as possible. The proposed solution considers the user's interaction context (i.e. combined user's, environment's and system's context) as well as the nature of the mathematical expression itself and of the user's preferences. The proposed solution is most efficient if the user is familiar with different presentation formats (e.g. Braille, speech, DotsPlus, EasyMath). Indeed, the proposed solution is apt for the given user's context and capacity.

Our future works involve the prototyping of this infrastructure and simulating its performance using several computing platforms. Such prototype will also be tested on visually-impaired users with varying interaction context.

Appendix A - Figures on Optimization Algorithms

```
Suitability Score of Modality Mj
Input:
1.  Interaction Context ICi.
2.  Modality Mj(j = 1 to card(power set(M)))
3.  Scenario repository SR
Output: Suitability score of Mj
Procedure:
1.  get ICi(i = 1 to max), cmax
2.  for each context (UC, EC and SC) do
3.    get maxw(w = 1 to cmax)
4.  endfor
5.  for each context ICiw do
6.    for each parameter of ICiw (ICParamwv) do
7.      P(ICi/Mj) = P(ICi/Mj) * P(ICParamwv/Mj)
8.    endfor
9.  endfor
10. P(Mj)= frequency(Mj)/cardinality(SR)
11. Suitability_Score_Mj = P(ICi/Mj)* P(Mj)
12. Return Suitability_Score_Mj
Endprocedure
```

Figure 12. Algorithm that calculates for the suitability score of each element M_j of the modality power set, given an interaction context IC_i.

Finding Optimal Modality $\hat{M}$
Input:
1. Interaction Context IC_i.
2. Modality set M
3. Scenario Repository SR

Output: Optimal modality $\hat{M}$
Procedure:
1. get IC_i and Modality set M
2. init = 1
3. MaxScore = Suitability Score Modality M_{init}
4. Optimal_modality = M_{init}
5. for each element M_j of power set of M do
6. if $(V_{in}$ OR $T_{in}) \in M_j$ AND $(V_{out}$ OR $T_{out}) \in M_j$ then
7.   if Suitability Score M_j > MaxScore then
8.   MaxScore = Suitability Score M_j
9.   Optimal_modality = M_j
10.   endif
11. endif
12. endfor
13. Return Optimal_modality

Endprocedure

Figure 13. Algorithm for finding the optimal modality.

Update Optimal Modality h_{best}
Input:
1. All available Media devices D
2. Optimal Modality h_{best}

Output: Optimal Modality h_{best}
Procedure:
1. for each modality M_i in h_{best}
2.   if media devices supporting M_i = failed then
3. M_i= Failed
4. h_{best}= h_{best}-M_i
5.   endif
6. endfor
7. if NOT((V_{in} OR T_{in}) AND (V_{out} OR T_{out})) then
8. h_{best}= Null
9. endif
10. Return h_{best}

Endprocedure

Figure 14. Algorithm for updating optimal modality based on the availability of media devices to support the chosen modalities

ACKNOWLEDGMENT

This research work has been made possible through the funding provided by the *Natural Sciences and Engineering Research Council of Canada* (NSERC), and the scholarship funds from the *National Bank of Canada* and the *Décanat des études* of *École de technologie supérieure* and *La bourse de cotutelle* provided by the governments of Quebec and France.

REFERENCES

[1] B. Stöger, K. Miesenberger, and M. Batusic, "Mathematical Working Environment for the Blind Motivation and Basic Ideas," in *ICCHP*: Springer, 2004, pp. 656-663.

[2] A. D. N. Edwards and R. D. Stevens, "A Multimodal Interface for Blind Mathematics Students," in *INSERM*, Paris, France, 1994, pp. 97 - 104.

[3] H. Cahill, C. Lineham, J. McCarthy, G. Bormans, and J. Engelen, "Ensuring Usability in MATHS," in *The European Context for Assistive Technology* Amsterdam: IOS Press, 1995, pp. 66-69.

[4] M. Preddy, J. Gardner, S. Sahyun, O. S. University, and D. Skrivanek, "Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math," in *International Conference on Technology and Persons with Disabilities*, Los Angeles, CA, USA, 1997.

[5] A. Podevin, "Accès aux formules mathématiques par des personnes non voyantes : étude et définition d'une méthode adaptée," Université de CAEN, 2002.

[6] H. Ferreira and D. Freitas, "Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project," in *ICCHP - 9th International Conference on Computer Helping People with Special Needs*, Paris, France, 2004, pp. 678-685.

[7] H. Ferreira and D. Freitas, "AudioMath: Towards Automatic Readings of Mathematical Expressions," in *Human-Computer Interaction International (HCII)*, Las Vegas, Nevada, USA, 2005.

[8] T. V. Raman, *Audio System for Technical Readings* vol. 1410. Berlin, Germany: Springer-Verlag, 1998.

[9] V. Moço and D. Archabault, "VICKIE: A Transcription Tool for Mathematical Braille," in *AAATE -Association for the Advancement of Assistive Technology in Europe*, Dublin, Ireland, 2003.

[10] F. Schwebel and R. Goiffon, "BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web," in *Journées nationales Caen*, Caen, France, 2005.

[11] P. Garlini and F. Fogarolo, "Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs," University of Padova, Padova, Italy 2003.

[12] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, Massachusetts: MIT Press, 2004.

[13] O. Kallenberg, *Foundations of Modern Probability*: Springer, 2002.

[14] T. Mitchell, "Machine Learning," McGraw-Hill, 1997.

[15] D. Lightfoot, *Formal Specification Using Z*, 2nd ed.: McMillan Press, 2001.

[16] G. Weiss, "Multiagent systems," *MIT-Press,* 1999.

[17] M. Wooldridge, *An Introduction to Multi-agent Systems*. Chichester, UK: Wiley, 2002.

[18] N. R. Jennings and M. J. Wooldridge, "Applications of Intelligent Agents," in *Agent Technology: Foundations, Applications, and Markets*, N. R. Jennings and M. J. Wooldridge, Eds.: Springer-Verlag: Heidelberg, Germany, 1998, pp. 3-28.

[19] A. Awde, M. D. Hina, C. Tadj, Y. Bellik, and A. Ramdane-Cherif, "A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users," in *The First International Conference on Grid and Pervasive Computing*, Tunghai University, Taichung, Taiwan, 2006.

[20] A. Awde, M. D. Hina, C. Tadj, A. Ramdane-Cherif, and Y. Bellik, "Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users," in *ISIE 2006, IEEE International Symposium on Industrial Electronics*, Montreal, Canada, 2006.

[21] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed.* San Francisco: Morgan Kaufmann, 2005.

Ali AWDE is currently a PhD student at the École de technologie supérieure (ÉTS). He obtained his master degree in Computer Science in 2003 from Université de Montréal. His research interests include multimodal multimedia for visually-impaired users, machine learning, and mathematics for the blind.
Email: *ali.awde.1@ens.etsmtl.ca.*

Manolo Dulva HINA is a student in a shared-PhD program between ÉTS, Canada and Université de Versailles–Saint-Quentin-en-Yvelines (UVSQ), France. He has a master's degree in Computer Science from Concordia University. His research interests include ubiquitous computing, autonomic computing, multimodal system and machine learning. Email: *manolo-dulva.hina.1@ens.etsmtl.ca.*

Chakib TADJ is a professor at ETS, Canada. He received his PhD degree from ENST Paris in 1995. His main research interests are automatic speech recognition, human-machine interface, multimodal and neuronal systems. Email: *ctadj@ele.etsmtl.ca.*

Yacine BELLIK is an assistant professor at LIMSI-CNRS (*Laboratoire d'informatique pour la mécanique et les sciences de l'ingénieur*). He holds a PhD and HDR in Computer Science. His research interests concern multimodal human-computer interaction, aid for the blind and ambient intelligence.
Email: *Yacine.Bellik@limsi.fr.*

Amar RAMDANE-CHERIF received his PhD degree from Université Pierre-et-Marie-Curie, Paris in 1998 in neural networks and AI optimization for robotic applications. He is currently a full professor in PRISM and LISV Laboratories, UVSQ in France. His research interests include software architecture, dynamic architecture, styles and quality attributes applied to robotic applications. Email: *rca@prism.uvsq.fr.*