Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users

Ali Awde, Yacine Bellik, and Chakib Tadj

***Abstract*—** Our adaptive multimodal system aims at correctly presenting a mathematical expression to visually impaired users. Given an interaction context (i.e. combination of user, environment and system resources) as well as the complexity of the expression itself and the user's preferences, the suitability scores of different presentation format are calculated. Unlike the current state-of-the art solutions, our approach takes into account the user's situation and not imposes a solution that is not suitable to his context and capacity. In this wok, we present our methodology for calculating the mathematical expression complexity and the results of our experiment. Finally, this paper discusses the concepts and principles applied on our system as well as their validation through cases studies. This work is our original contribution to an ongoing research to make informatics more accessible to handicapped users.

***Keywords*—** Adaptive system, intelligent multi-agent system, mathematics for visually-impaired users.

I. Introduction

An adaptive computing system is a system that is capable to adapt its behavior according to changes in its environment or on user's situation or in parts of the system itself. Hence, the system can reason and react with no or very little human intervention.

A multimodal system allows user to interact with more than one mode of interaction. Indeed, incorporating *multimodality* into a computing system makes it more accessible to a wider range of users, including those with impairments. With multimodality, the strength or weakness of a modality or media device is decided based on its suitability to a given context. For example, visual modality (screen) is not appropriate to visually impaired users hence tactile modality (Braille terminal) or speech or both (using many media concurrently) can replace it.

Making mathematics accessible to visually impaired people is a significant challenge, due to the following reasons: First, the visual mathematical representation 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 (e.g. Braille) requires further information to explain a term (for example, an exponent) so the blind users can understand the expression correctly. Also, the conversion into an audio format is often ambiguous. Third, the vocabulary terms used by sighted people are quite large as compared to the amount of data that can be made accessible to blind people. For example, traditional Braille[1] utilizes 6-dot character, which allows 64 possible characters. This quantity of characters is not enough to represent all frequently-used mathematical symbols (e.g. "X" is represented by a sequence of two characters: ",x"). Also, large quantity of symbols is a challenge to a blind user.

The current state-of-the art proposes systems and solutions that translate mathematical expressions into other presentation formats. For instance, some systems convert a standard algebraic expression into its audio equivalent, while others convert a mathematical document into its Braille representation. Yet others produce mathematical expression's equivalent and present them as printed document in which characters are embossed and meant to be touched.

Our work has been conceptualized after having reviewed the current state-of-the-art systems and solutions and found out that most, if not all, of available systems for presentation of mathematical expressions to visually-impaired users do not provide the user with the desired autonomy; that most of these systems do not take into consideration the current interaction context in their system's configuration and that the prevailing systems provide only one choice of presenting mathematical expressions, meaning that they are not multimodal. In other word, our work addresses the weaknesses that we found in the current state-of-the-art systems and solutions.

In this research, we take into account the current interaction context and the nature of expression because we have noted that the choice of presentation format varied also based on the nature of the expression. Various previous works [1] treated the interaction context. In this paper, we concentrate on studying the complexity of an expression. We demonstrate here that the parameters that constitute expression context

A. Awde is with LATIS laboratory, Département de Génie Électrique, Université du Québec, École de technologie supérieure. 1100, rue Notre-Dame, Montréal, Qc, H3C 1K3, Canada. E-mail: ali.awde.1@ens.etsmtl.ca.

Y. Bellik is with LIMSI-CNRS laboratory, Université Paris-Sud, B. P. 133, 91403 Orsay, France. E-mail: yacine.bellik@limsi.fr.

C. Tadj is with LATIS laboratory, Département de Génie Électrique, Université du Québec, École de technologie supérieure. 1100, rue Notre-Dame, Montréal, Qc, H3C 1K3, Canada. E-mail: ctadj@ele.etsmtl.ca

[1] http://6dotbraille.com

cannot be fixed or pre-defined. They have to be learned based on many parameters. In this regard, our approach is based on observations and a mathematical model to define the expression complexity. Once the complexity is calculated, it can be integrated to other contextual information in order to find the most appropriate presentation format for each specific visually impaired user. This last step is not addressed in this paper.

In this paper, we present the infrastructure of a multimodal computing system that presents mathematical expressions to visually-impaired users by taking into account his interaction context, preferences and the nature of the expression itself. The rest of this paper is structured as follows. Section II presents related researches and highlights the novelty of our work; Section III presents the purpose, objectives and questions of the work. Section IV defines media, modality and presentation format. Section V presents the design of our system and the contextual information. In section VI, we present expression components and a mathematical taxonomy. Section VII presents experiment protocol and running experiment. In section VIII, we present and argue the results of the experiment. In section IX, we apply the concepts experimented in the previous section. Finally, we present our future works and conclusion in section X.

II. Related Work

To a visually-impaired user, understanding mathematical expression requires repeated passages over the expression, sometimes skipping some secondary information, only to reverse back to it again and again until the user fully grasps the expression. A complicated task like this is detailed in [2]. Some tools, however, have been developed to lessen the complexity of performing a similar task, among them being Mathtalk [3], Maths [4], DotsPlus [5], EasyMath [6] and AudioMath [7, 8]. In Maths, and MathTalk the user can read, write and manipulate mathematics using a multimedia interface containing speech, Braille and audio. VICKIE (*Visually Impaired Children Kit for Inclusive Education*) [9] and BraMaNet (*Braille Mathématique sur InterNet*) [10] are transcription tools that convert mathematical document (written in LaTex, MathML, HTML, etc.) to its French Braille representation. Labradoor (*LaTex-to-Braille-Door*) [11] converts an expression in LaTex into its Braille equivalent using the Marburg code. MAVIS (*Mathematics Accessible To Visually Impaired Students*) [12] supports LaTex to Braille translation using Nemth code. DotsPlus is a tactile method of printing documents that incorporates both Braille and graphic symbols (e.g. $\prod$, $\sum$, etc.). For EasyMath [6], the 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.

As a background, some special Braille notations have been developed for mathematics. Also, different Braille code notations for mathematics are available depending on the country. Some of these codes are the Nemeth Math code [13] which is used in the USA, Canada, New Zealand, Greece and India; the Marburg code [14] used in Germany and Austria, the French Math code [15].

None of the tools cited above, however, is complete. Studies were conducted evaluating these tools based on users' needs [6, 16]. 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 tool has its own set of limitations. For example, Aster (*Audio System for Technical Readings*) [17] uses only a LaTex document, and AudioMath uses only a MathML document, in its conversion to its equivalent vocal output. Our approach, therefore, is to get the strength of each tool, integrate each one of them into our work in order to build a system that (1) broadens the limits of utilization, (2) provides the user with opportunities to access as many document types as possible, and (3) presents data output in as many suitable formats as possible after considering user situation and the expression complexity. This work is an essential contribution because we offer all types of data presentation formats yet the system requires minimum explicit intervention from the user.

Also we noted that although functional, the system's effectiveness is limited as the modalities for user interaction are already pre-defined. In contrast, a computing system becomes more flexible if no pre-defined input-output modalities are set. In fact, the output presentation of information should be based on the user's application and interaction context (user, system, and environment) which could possibly be in constant evolution. The framework for intelligent multimodal presentation of information [18] is an example of such flexible system. The system's user interface also should be adaptive to these context variations while preserving its usability. Demeure's work [19] exhibits plasticity in context adaptation. Indeed, the forms of modality should be chosen only based on their merits to a user's interaction context. This is the approach adopted in our work.

In [20], the use of multimodal interaction for non-visual application have been demonstrated. The strength of multimodality lies in the selection of a modality over another after having determined its suitability for data input or output given a specific situation. To visually-impaired users, multimodality is even more important as it provides them greater opportunities to use informatics. In determining the appropriate modality, the user situation plays an important role. In our work, we expanded the notion of user context as one that includes additional handicaps and the user preferences on the priority rankings of media devices and presentation formats.

III. Purpose of The Work

The main purpose of this paper is to provide computing infrastructure to visually-impaired users through multimodality. This is in-line with the need for improving the quality of existing applications and for providing visually-impaired users autonomy as they still need assistance of sighted people.

There are two main parts in selecting the suitable presentation format: the interaction context and the expression nature. The interaction context is not our focus here (see [21]) so we present it briefly. However, we present our methodology for calculating the expression complexity based on the experiments. Also, we present an architectural framework of an adaptive system that can reason and react accordingly with no or very little human intervention.

We have been working on adaptive multimodal systems ensuring access to mathematics for visually impaired users. The objectives of our works are: a) to improve the quality of applications that present mathematics to visually impaired users, b) to come up with an effective and intelligent way of presenting mathematics so that users can master the application without the assistance of sighted people and c) to provide various suitable presentations format to users.

In this paper, our immediate objectives are a) to provide an efficient method to determine the expression complexity and b) to provide a methodology to integrate the complexity in an adaptive multimodal system.

In order to accomplish these objectives, we prepared the following questions:

- What are the factors (i.e. modalities, media, context...) that applications need to consider in order to produce high quality application which present mathematics to blind users?
- Why are the currently available tools not so effective?
- Which factors and parameters determine the complexity of a mathematical expression?
- How the complexity approach introduced in this study can be integrated in the system to select the most suitable presentation format?

IV. Modality, Media and Presentation Format

A. *Modality and Media*

In our work, we adopt the notions of media and modality that are defined by Bellik in [22].

1. *Modality* is defined by the information structure as it is perceived by the user (e.g. text, speech, Braille, image, etc.).

2. *Media* is defined as a device used to acquire or deliver information or data (e.g. screen, terminal Braille, mouse, keyboard, etc.)

Here, *Vocal* and *Tactile* modalities are possible since we address visually impaired users. Also, in general, interaction 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}} wherein V_{in} = *vocal input,* V_{out} = *vocal output,* T_{in} = *tactile input and* T_{out} = *tactile output* then interaction is possible under the following condition:

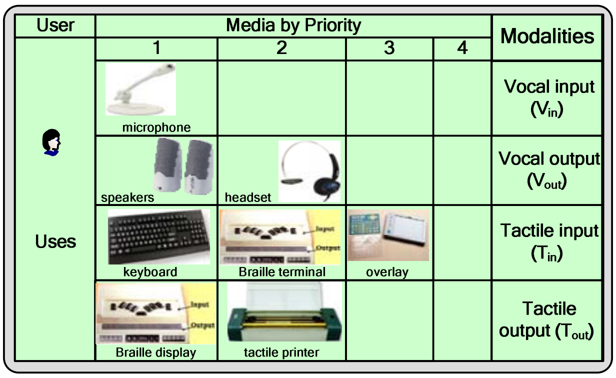
$$Interaction\ Possible = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \quad (1)$$

where the symbols ∧ and ∨ denote logical AND and OR, respectively.

There are usually more than one media that support a specific modality (see Fig. 1). For example, a regular keyboard, an overlay keyboard and a Braille terminal are all devices supporting tactile input modality. Activating them all is plain redundancy; hence the system must select only the top-ranked device.

A. *Presentation Format*

MathML 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. *MathML* is used to edit mathematical expressions. Also, LaTex is used to do the same task. These tools present mathematical expression in a bi-dimensional format which is not appropriate for visually impaired users. There are some alternative formats (i.e. Braille, Audio, DotsPlus, EasyMath, etc.) that are addressed to this kind of users. For example, Fig. 2 shows two specimens expression (in bi-dimensional form) and their equivalents representation in DotsPlus and Braille.



User	Media by Priority				Modalities
	1	2	3	4	
Uses	microphone				Vocal input (V_{in})
	speakers	headset			Vocal output (V_{out})
	keyboard	Braille terminal	overlay		Tactile input (T_{in})
	Braille display	tactile printer			Tactile output (T_{out})

Fig. 1 A sample of media devices priority table and different modality's forms.

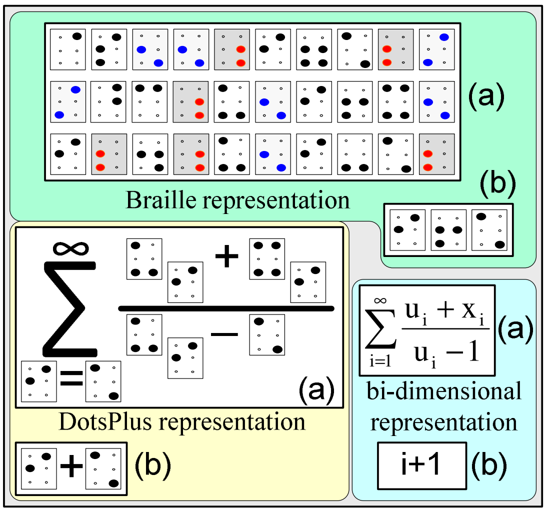

(a)
(b)
Braille representation
+
−
=
(a)
DotsPlus representation
+
(b)
$\sum_{i=1}^{\infty} \frac{u_i + x_i}{u_i - 1}$
(a)
bi-dimensional representation
i+1
(b)

Fig. 2 Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format.

The presentation formats available in the user's computing system are all inputs to the presentation format selection process. Based on interaction context, available media, modalities, expression complexity and user's preferences, the system selects optimal presentation format. For this paper, selection of appropriate presentation format is not discussed (see [23] for details).

V. Adaptive Multimodal Computing Design

In this section, we provide the infrastructure that satisfies the design specifications of our adaptive multimodal system.

A. *Architectural Framework*

Fig. 3 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 prevent the

International Science Index, Mathematical and Computational Sciences Vol:2, No:3, 2008 waset.org/Publication/11248

possibility of ripple effect from propagating to other system components. The various layers and their functionalities are as follows:

1. *Physical Layer* – contains all the physical entities of the system, including devices and sensors. The raw data from this layer are sampled and interpreted and forms the current instance of interaction context.
2. *Context Gathering Layer* – detects current interaction context;
3. *Control and Monitoring Layer* – controls the system, coordinates the detection of interaction context, the mathematical expression, its presentation and/or manipulation;
4. *Data Analysis Layer* – here, the presentation format of the mathematical expression is selected based on available resources and user's context;
5. *Data Access Layer* – allows search/edit of mathematical expression;
6. *Presentation Layer* – presents mathematical expression via optimal presentation format.

An agent programming technique was chosen because the traditional techniques (i.e. functional or object-oriented programming) are inadequate in developing tools that react to environment events. Multi-agent systems (MAS) [24, 25] 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 [26]. Some works on MAS for visually-impaired users are exposed in [20, 27]. In this work, however, our concern is on the correct representation of a mathematical expression while providing users autonomy. A multi-agent system is used to implement the architectural framework of Fig. 3. The functionalities of the various agents in our multimodal system are shown in Table I.

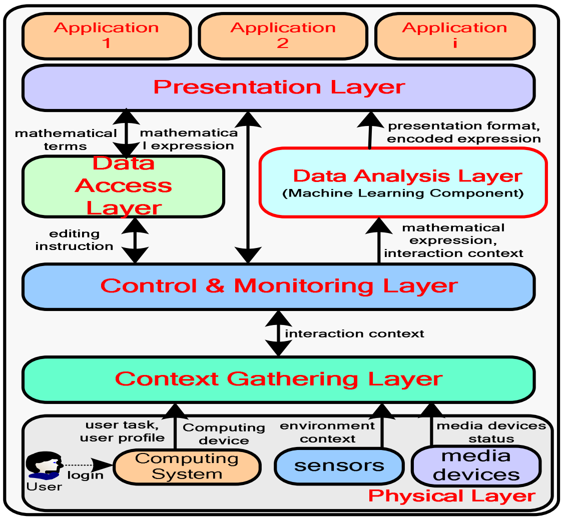


Fig. 3 Architectural layer view of our multimodal computing system for visually-impaired users.

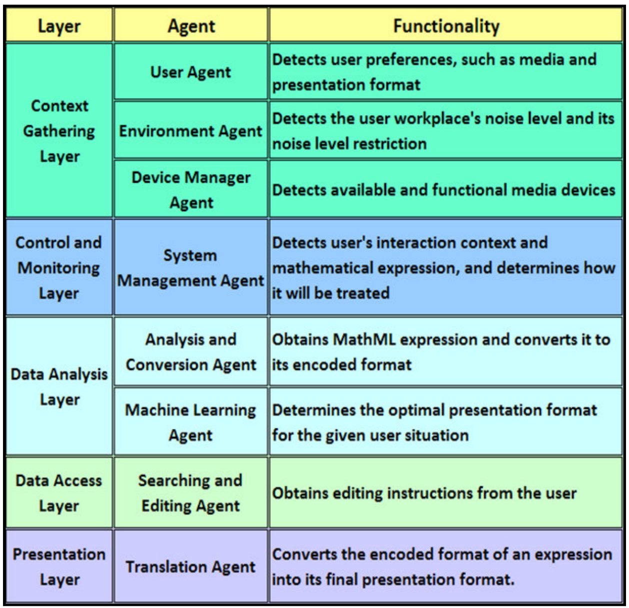
B. Contextual Information

Our approach is based on analysis of contextual information and learning techniques to design a system that is capable of adapting to the context (i.e. *interaction context (IC)* and *complexity of expression*). For this paper, interaction context is not discussed in detail since it is not related to this paper's content. IC is formed by combining the contexts of the *user*, his *environment*, and his computing *system.*

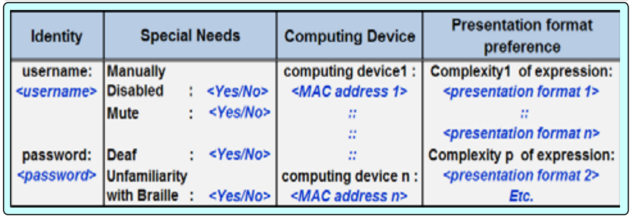
i. The user context

In this work, the user context (UC) is a function of user profile (including any handicap) and preferences. A sample of user profile, in generic format, is shown in Fig. 4. 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 I
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 presentation 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	Presentation format preference
username: *<username>*	Manually Disabled : *<Yes/No>* Mute : *<Yes/No>*	computing device1 : *<MAC address 1>* :: ::	Complexity1 of expression: *<presentation format 1>* :: *<presentation format n>*
password: *<password>*	Deaf : *<Yes/No>* Unfamiliarity with Braille : *<Yes/No>*	:: computing device n : *<MAC address n>*	Complexity p of expression: *<presentation format 2>* *Etc.*

Fig. 4 A sample of user profile.

ii. The environment context

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. EC affects the selection of media hence the choice of suitable format. For example: in a library where silence is required, sound-producing media (e.g. speaker) needs to be muted or deactivated.

iii. The system context

In our work, the *system context* (**SC**) represents the user's computing device and the available media devices. The computing device (e.g. PC, laptop, PDA, cellular phone) also affects the modality selection. For example, using a PDA or

International Science Index, Mathematical and Computational Sciences Vol:2, No:3, 2008 waset.org/Publication/11248

cell phone prevents the user from using tactile output modality so Braille format is not possible.

iv. The expression complexity

The complexity of the expression affects the choice of the format of presentation. In case of simple expressions (see Fig. 2, expression (b)), the user will choose simple presentation format such as Braille or audio. Note that when the expression is complex, user has to choose more complex presentation format such as DotsPlus's presentation (e.g. expression (a) in Fig. 2). Hence, the complexity of the expression is important for determining the suitable presentation format. A mathematical expression consists of one or more operands and operators. Operators are symbols that represent a function, whereas operands are its arguments. Therefore we developed a method based on the syntax tree of the mathematical expression to determine its complexity.

In our work, we retain the three following parameters: the depth of syntax tree, number of operands and operators. We also examined the impact of mathematical expression branch (e.g. algebra, analysis, etc.) on its complexity. In the next sections, we provide the details of our approach to calculate the expression complexity.

VI. Mathematics

A. Mathematical Taxonomy

In the literature, mathematics are defined as the study of the measurement, properties, sets, shapes and motions of physical objects using numbers and symbols. Also we know it as a science of modeling, demonstration and calculation. In our work, we divide mathematics into 6 branches: algebra, arithmetic, calculus, geometry, logic and Statistics & Probability.

B. Principal Branches of Mathematics:

i. Arithmetic: This branch is also known as numbers theory. It deals with numbers (natural or relative) and their proprieties. Arithmetic does not consider the literal operands that are variables. Our system covers all the arithmetic expressions that contain symbols such as +, -, ×, ÷, /, Σ, Π, exponent.

Example: $1+3\times5=16$

ii. Algebra: It is the branch that substitutes letters for numbers. Algebra deals with symbols and variables. Thus we can consider it as an extension of arithmetic where letters, symbols represent variables. Also, this branch is about the study of quantity, relation and structure (e.g. set, group). In this work, we focus our experience on expressions made of operations and functions such as the four elementary arithmetic operators (+,-,*,/).

Example: $4x^3 -2x^2 +x-3=0$

iii. Analysis: It is a branch which deals with numbers (real, complex) and their relations. It studies continuously changing quantities and relations which assigns to a variable another corresponding variable. The functions can be presented graphically (curve). Our topics cover the functional expressions such as algebraic and differential equations, variables and functions. On the other hand, it does not manage curve and figures.

Example: $\boldsymbol{f(x)=\frac{3x+5}{\sqrt{2x}}}$

iv. Geometry: It is the science of the figures in the plan and the forms (or volumes) in space. Trigonometry is part of the geometry. It makes it possible to study angles and their properties. In our research, we do not treat the figures as well as the vectors and their properties. However, we treat trigonometric expressions including functions such as sin, cos, tan.

Example: $\cos^2(x)+\sin^2(x)=1$

v. Logic: it is the study for composing proofs, which gives us reliable confirmation of the truth of the proven proposition. It is also called "science of the reasoning". It is used to approve a reasoning, to distinguish the truth of the forgery. This branch is widely used in computer science field. The logic expressions we are considering here are propositional logic and first order logic.

Example: $a \wedge b$

vi. Statistics & Probability: Probability is the likelihood or chance that something is the case or will happen. Probability is essential to many human activities that involve quantitative analysis of large sets of data. Statistics is a mathematical science about the collection, description, analysis, interpretation of data and prediction. In this branch, we consider expressions written to describe a phenomenon but we do not treat graphs or histograms.

Example: $P(x)=½$

C. Mathematical Expression

A mathematical expression consists of several operands connected by operators. In a mathematical expression, brackets and parentheses group terms aim to simplify reading of expression and specify priority of calculations. Each set of terms ranging between two brackets (open and close) is called block.

i. Operands

An operand is defined as the element on which operations are carried. It can be constant or variable. In a mathematical expression, there is always at least one operand. The nature of the operand varies from a mathematical branch to another. For example, in arithmetic the operands are numbers. However, in algebra they are numbers and literal variables, and in logic, the operands take truth values to make predicates. Operand can be simple (single operand) or composed (a group of operand and operator enclosed in parentheses).

ii. Operators

An operator performs a logical or mathematical operation on the operands in an expression. It settles the relation between terms of the mathematical expression. Each mathematical branch has some specific operators although there are common symbol operators between these branches. The complexity of an operator depends on the number of its operands. Indeed, there are several types of operators:

- Unary operator: it needs just only one operand such as the square root, sin, logical predicates (NOT).
- Binary operator: it involves exactly two operands such as the addition (+), the multiplication (*), logical predicates (e.g. OR, AND), etc.
- N-ary operators: when the operator has three operands or more, it is known as n-ary operator. For example, definite

integral has 3 arguments (i.e. *f(x).dx*, *upper and lower* limits).

In an expression that contains multiple operators, there are rules to decide the order in which operators are evaluated. These rules are called operator priority. Operators of higher priority within an expression are executed before operators of a lower priority. For example multiplication has a higher priority than addition. To increase the priority of an operator, we place it and its operands inside parentheses. Also, these parentheses or brackets contribute to read and understand the expression easily in dividing the expression to subsets.

iii. The syntax tree

To determine the complexity of a mathematical expression, it is necessary to consider all expression components. Operators and operands are easily detected while reading the expression. However, presence of many brackets and especially overlapping brackets increase the challenge and complicates analysis of the expression. Therefore, we elaborated a solution based on the principle of syntax tree of the expression where internal nodes are labeled by operators (nonterminal), and the leaf nodes represent the operands of the operators (terminal). Thus, the leaf nodes only represent variables or constants [28].

Finally we consider *the depth* of tree being the number of connections that we encounter starting from root node (i.e. topmost node) to the farthest leaf. This parameter points to the number of groups (i.e. parentheses or brackets) in the expression. In this work, the tree of an expression has a unique root and at least one leaf.

To build a syntax tree of the expression, we proceed in a similar way as the conversion of the expression into postfix form. We build subtrees for the subexpressions (i.e group of terms between 2 brackets) by creating a node for each operator and operands. The children of an operator node are the parents of the nodes representing the subexpressions constituting the operands of that operator. The subexpressions are determined based on the nature of operators and their priorities. When operators have identical priority, the construction of subexpressions is done by grouping operators and operands from left to right (i.e. left-associative or the order of evaluation).

Consider, for example, this expression:

$$\frac{(1+x)(y+5\times x)}{x+2+y}$$

In this expression, parentheses in numerator identify well the two operands of the multiplication (2 subexpressions). However for the term (y+5×x), the multiplication has a higher priority than the addition. In this case, multiplication operator and its operands (i.e. 5 and x) become a subexpression and addition operator with its operands (i.e. y and 5×x) are another one. On the other hand, in the denominator, there are 2 identical operators. Therefore we divide denominator in 2 subexpressions according to the order of left associative. So the expression can also be written as follows:

$$\frac{(1+x)(y+(5\times x))}{((x+2)+y)}$$

Fig. 5 shows the syntax tree of that expression. Note that left side of the figure presents the numerator and right side illustrates the denominator.

The root node represents the fraction bar. Internal nodes represent operators in the expression (*i.e.* ×,+,+,×,/,+,+). While leafs correspond to variables and constants (*i.e. 1, x, y, 5, x, x, 2, y*). For the depth, we count 4 connections from the root (i.e. /) to the farthest leaf (*i.e. 5 or x*).

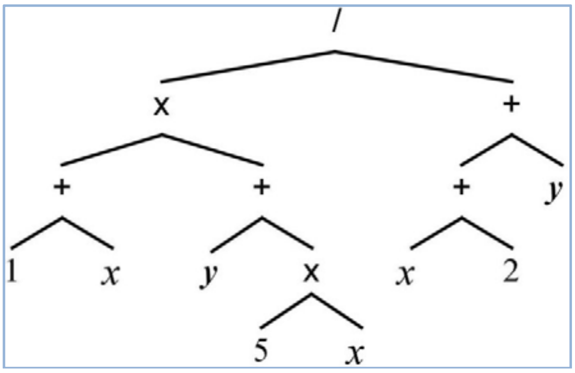


Fig. 5 A syntax tree of the sample expression.

VII. The Experiment

A. Hypothesis

In our experiment, we have some assumptions:

- There is a relation between the three parameters of the expression (operators, operands, depth) and its complexity. Then our tentative is to determine the function which allows calculating the complexity.
- The parameters of the expression have not the same influence on the complexity so we have to determine the weight of each one of them.
- The branch of the expression may affect its complexity. In probability branch, special symbols such as (∑, ∏, etc.) are used often and they are more complex than arithmetic symbol such as +,-, ×, etc.

B. Evaluation Protocol

Our objective of the evaluation is to establish a relation between the three parameters (operators, operands, depth) and the complexity of the expression. According to the mathematical taxonomy that we presented above, mathematics comprise 6 branches. We composed ten expressions of each branch, with various complexities to cover all possible degrees of complexity (varying the number of operators, operands and parentheses from one expression to another).

We want to check if the complexity of the mathematical expression varies from a branch to another when both have the same number of parameters as well as the relation between the type of mathematical operators and expression complexity. Also, we made another test where the collection contains expressions of various mathematics branches.

In our experience, the main hint to the complexity is the time allocated by participant for reading and memorizing the expression. Also, error rate during the rewriting of the expression gives another indication to the level of complexity.

To measure the error rate, we took a metric that is similar to a common metric of the performance of a speech recognition system and machine translation system. That metric is known as word error rate (WER) [29] . It is derived from Levenstein distance. The distance between two sentences is the minimum number of insertions, deletions and substitutions required to transform one sentence into the other. Here, we assign the same weight to these three operations. Also, we use expressions instead of sentences and terms (operator and operand) instead of words. The WER is the distance between an expression reference and its corresponding expression written by the participant. Formally, the WER is defined as:

$$WER = \frac{\mathrm{I} + \mathrm{D} + \mathrm{S}}{\mathrm{N}}$$

Where I is the number of the insertions, D is the number of the deletions, S is the number of substitutions and N is the number of operators and operands in the expression reference.

C. Experiment Running

The participant must take his time to read and memorize the expression shown on-screen. When participant think that he knows well the expression, he hide it and try to write the expression on a sheet. At the end, the participant assigns a digit (from 1 to 5) to the expression to indicate its complexity according to his experience and values of Table II. In this experiment, participant considers the expression that takes a long time to be memorized as very complex.

TABLE II
THE COMPLEXITIES AND ITS SUITING VALUES.

Complexity	Value
Very Simple	1
Simple	2
Average	3
Complex	4
Very Complex	5

The purpose of the experiment is to verify if the participants have the same tendency to classify expressions or the order varies from one person to another. In this way, we can determine the relationship between the profile of the reader (i.e. participant) and the complexity of the mathematical expression. The expressions are presented to the participants in a random order of complexity.

The expressions are presented on a screen of a computer in order to measure the time for reading, memorizing and rewriting the expression. However, the rewriting of expressions is made by hand. The participants in this experiment are university graduates in different studies, such as computer science, mathematics, chemistry, literature, etc. We choose this degree of studies because the types of the operators used in this experiment are not recognized by college students. The participants are skilful to use computer therefore we did not encounter problems of this kind during tests. There were 50 participants. A test of ten expressions takes in average 15 minutes. After each test passed by the participant, we collected information by noting times taken for each expression, and by checking the syntax of the expression written by this participant. The error rate for each expression is then calculated.

VIII. RESULTS OF EXPERIMENTS AND DISCUSSIONS

In this section, we present for each branch of mathematics the data collected during our experiments and the corresponding results. Results are shown in the following tables and graphs. The first column in these tables presents the mathematical expression number, the second column shows the number of operands, the third is the number of operators and the fourth is the depth of the syntax tree of the expression. These data are easily extracted from the syntax tree of the expression. The last four columns of the tables are results of our experiments. We calculated the average time taken by all participants for each expression. The rate column is the average of the error rates of all the participants. Complexity row shows in which complexity level the participant has placed the expression (there are 5 levels).

On graphs, R. Time and W. Time mean average time taken by all participants to read and write an expression respectively.

A. Arithmetic

Table III presents the results for arithmetic expressions. The number of operands varies between 1 and 8. The number of operators varies between 2 and 9. Here, the depth of the syntax tree is not more than 4 as the depth is always less than the number of operators.

Let us take expressions 2 and 4 (see appendix). They have the same number of operands "5" and operators "4" but their tree depths are different ("3" and "4"). The average time for reading and memorizing expression 2 was 13.06 seconds. On the other hand, expression 4 took almost 24.87 seconds. In the same way, the error rate passed from 1.48% to 8.15%, and the majority of the participants gave complexity 4 and 3 to expression 4 and 2 respectively. We can say that it is due to the fact that expression 4 includes several brackets. Thus, the depth has an influence on the complexity of the expression and should be taken it in to account it in the other branches of mathematics.

TABLE III
EXPERIMENT'S RESULTS OF ARITHMETIC BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	3	2	2	5.01	6.67	0.00	1.64
2	5	4	3	13.06	11.47	1.48	2.67
3	4	3	3	7.27	4.40	0.00	1.81
4	5	4	4	24.87	16.73	8.15	4.33
5	4	4	4	10.27	8.20	1.90	2.53
6	8	7	3	23.47	10.67	11.11	3.73
7	4	3	3	9.13	5.87	0.00	1.87
8	6	9	4	20.00	13.80	1.33	4.04
9	7	7	4	28.60	12.20	19.05	4.67
10	1	2	2	2.20	3.73	0.00	1.01

Fig. 6 illustrates reading time that reaches its high with expressions 4, 6, 8 and 9. These expressions have the most operators, operands and depth (i.e. 4).

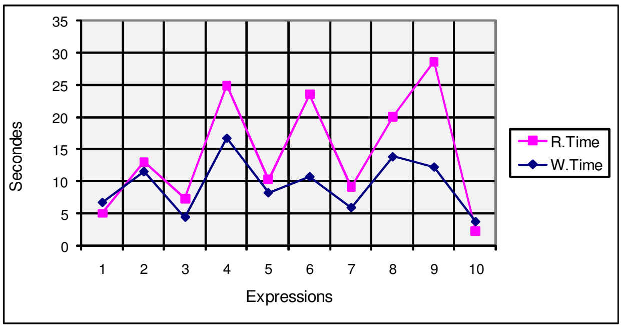


Fig. 6 Curves representing reading and writing times taken by participants during experiment of arithmetic branch.

B. Algebra

Table IV and Fig. 7 represent test results for algebra branch. We notice that the tendencies of participants continue in the same direction to confirm our assumption that the complexity of mathematical expressions depends on these three parameters (i.e. operators, operands, depth). For example, expression 7 has higher parameters than other expressions. Experiments classify this expression as the most complex expression among all expressions in this branch.

In Fig. 7, we notice that expression 7 takes the highest reading time (26.82 seconds).

TABLE IV
EXPERIMENT'S RESULTS OF ALGEBRA BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	4	4	4	4.80	7.13	0.00	1.99
2	6	7	4	17.62	12.07	1.54	3.69
3	7	6	4	8.73	7.62	9.23	3.62
4	5	6	4	12.93	10.02	6.06	3.40
5	3	2	2	2.20	2.18	0.00	1.52
6	9	8	4	12.61	11.13	5.49	3.87
7	8	11	5	26.82	16.20	15.19	4.93
8	4	3	2	4.92	4.62	0.00	1.29
9	2	1	1	2.13	4.73	0.00	1.07
10	5	5	3	6.64	7.87	12.67	2.87

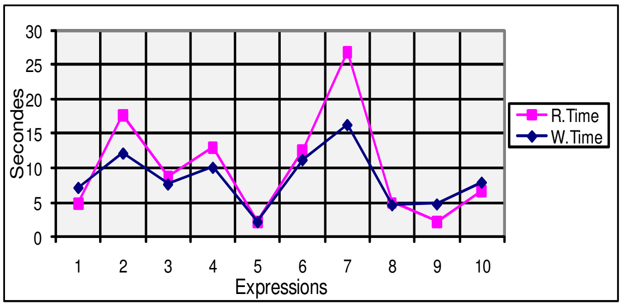


Fig. 7 Curve representing reading and writing times taken by participants during experiment of algebra branch.

C. Analysis

Let us take the two expressions, 6 and 9, which are considered complex by participants. As shown in Table V, parameters of expression 6 are higher than the ones of expression 9. Then, expression 6 must take more time and should be considered the most complex among all expressions in this experiment. However, the experimental results illustrated in Fig. 8 shows the contrary (20.53 seconds for expression 6 and 22.94 for expression 9). In fact, there are some participants not familiar with the mathematical function "**log**", and then reading time increased for this expression.

A. Geometry / Trigonometry

In this branch, expressions cover all trigonometry functions. Test results are presented in Table VI and Fig. 9.

Expressions 2 and 9 have the same number of operands, but expression 9 has one more operator. However, the complexity of the expression 9 is higher than the expression 2. We asked participants about that difference. They think that it is because of the presence of numbers in the expression. They also found that expression 2 is similar to arithmetic expression. In addition, for scientific participants, these 2 expressions had the same complexity, but for other participants the expression 9 was more complicated.

TABLE V
EXPERIMENT'S RESULTS OF ANALYZE BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	5	5	3	8.45	7.53	0.00	2.56
2	2	2	2	2.80	3.31	0.00	1.09
3	3	1	1	2.82	5.04	0.00	1.01
4	4	3	3	4.87	4.35	0.00	1.72
5	6	7	4	14.61	11.42	18.97	3.97
6	7	10	5	20.53	16.41	5.88	4.53
7	4	4	3	14.07	5.82	4.17	2.80
8	8	7	3	12.73	11.67	4.44	3.93
9	5	8	5	22.94	13.13	9.23	4.67
10	7	6	4	17.02	11.92	7.18	3.59

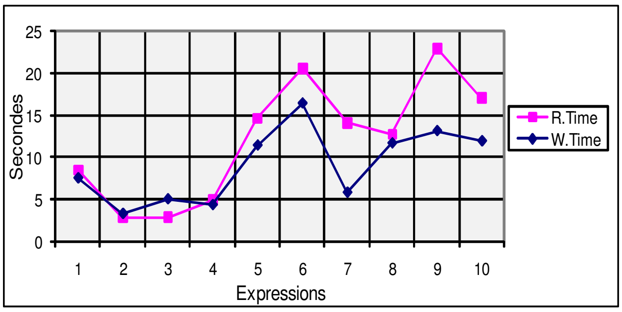


Fig. 8 Curves representing reading and writing times taken by participants during experiment of analysis branch.

TABLE VI
EXPERIMENT'S RESULTS OF TRIGONOMETRY BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	3	4	3	6.47	6.21	0.00	1.51
2	5	5	4	15.36	7.87	0.00	2.53
3	6	7	4	19.53	9.32	1.54	3.64
4	4	6	4	17.51	8.24	1.34	3.13
5	2	5	3	8.34	9.72	4.76	2.04
6	1	3	3	6.72	4.28	0.00	1.13
7	7	12	5	37.73	16.42	1.41	4.61
8	8	9	5	24.92	12.26	9.42	4.58
9	5	6	5	22.27	9.04	6.06	4.24
10	7	8	5	30.40	12.82	12.02	4.47

Graph in Fig. 9 shows that expressions 7, 8, 9 and 10 are the most complex among all expressions, and this is due to the largest number of parameters of these expressions.

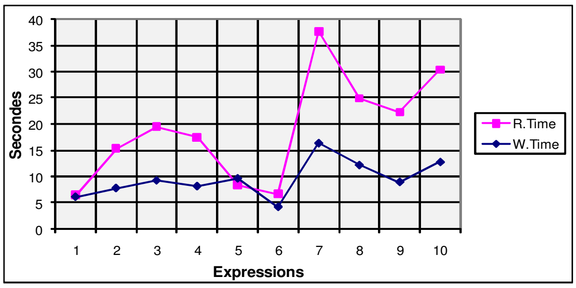


Fig. 9 Curves representing reading and writing times taken by participants during experiment of trigonometry branch.

B. Logic

During our tests, we prepared a summary of this branch to explain it to participants who had not enough knowledge in this field.

Even with this assistance, we can notice, in Table VII and Fig. 10, that participants took longer time to read logic expressions comparing to other branches like arithmetic even with the same number of operands and operators.

In the same way, expressions with a higher number of operators and operands need a longer reading time. For example, expression 8 took more than one minute for reading. This is due to the high number of terms.

TABLE VII
EXPERIMENT'S RESULTS OF LOGIC BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	**4**	**2**	**2**	**9.40**	**6.46**	**0.00**	**1.92**
2	**8**	**6**	**4**	**21.73**	**17.81**	**1.91**	**2.87**
3	**2**	**3**	**3**	**10.52**	**5.88**	**0.00**	**1.59**
4	**7**	**6**	**3**	**26.61**	**20.32**	**0.00**	**4.15**
5	**6**	**7**	**3**	**22.24**	**18.53**	**1.03**	**3.94**
6	**3**	**4**	**3**	**19.07**	**11.34**	**13.31**	**2.48**
7	**5**	**5**	**3**	**22.62**	**16.41**	**3.34**	**3.46**
8	**9**	**10**	**5**	**61.53**	**28.12**	**27.72**	**4.92**
9	**5**	**8**	**6**	**37.12**	**32.47**	**22.56**	**4.71**
10	**6**	**7**	**4**	**26.34**	**15.21**	**6.15**	**3.52**

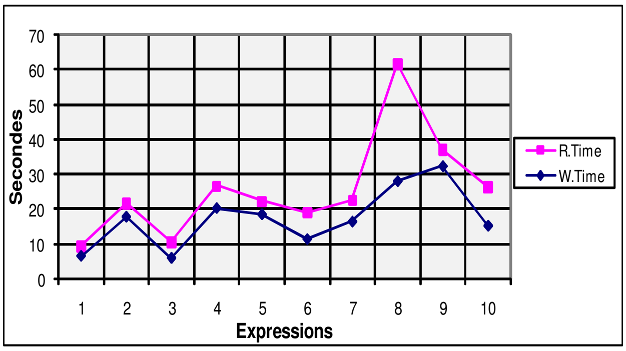


Fig. 10 Curves representing reading and writing times taken by participants during experiment of logic branch.

C. Statistics & Probability

In this mathematical branch, we use often operators which are "tri-nary" such as the summation. This will increase the quantity of information to be memorized and therefore the reading time. This hypothesis was confirmed during evaluations, as shown in Table VIII and Fig. 11.

For example, participants took on average of 56.72 seconds to memorize expression 10. This time is justified by the appearance, in the expression, of a rarely used function (i.e. VAR) and the large number of parameters compared to other expressions.

A. Combination

In this section, the set of expressions is a collection of 12 expressions of all mathematical branches. Here, our objective is to verify experiment results when expressions are a combination of various mathematical branches. To do so, the same participants of previous experiments realized this test. Table IX and Fig. 12 show the test results.

TABLE VIII
EXPERIMENT'S RESULTS OF STATISTICS & PROBABILITY BRANCH.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	**7**	**9**	**5**	**40.42**	**18.52**	**18.33**	**4.28**
2	**5**	**4**	**3**	**22.23**	**15.47**	**11.11**	**2.64**
3	**3**	**1**	**1**	**5.87**	**5.41**	**0.00**	**1.04**
4	**9**	**7**	**5**	**33.87**	**23.02**	**7.08**	**3.85**
5	**2**	**1**	**1**	**2.03**	**3.52**	**0.00**	**1.02**
6	**4**	**6**	**4**	**18.81**	**11.46**	**4.04**	**2.53**
7	**6**	**5**	**5**	**35.04**	**19.81**	**5.15**	**3.87**
8	**6**	**3**	**3**	**24.93**	**12.72**	**5.19**	**3.33**
9	**4**	**2**	**2**	**9.41**	**7.03**	**0.00**	**1.53**
10	**8**	**12**	**6**	**56.72**	**21.53**	**14.68**	**4.83**

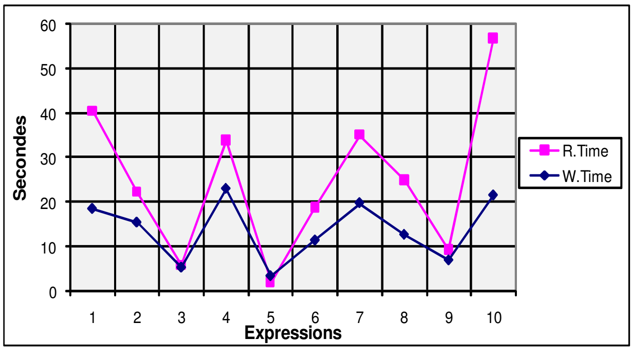


Fig. 11 Curves representing reading and writing times taken by participants during experiment of probability and statistics branch.

The results of this test confirmed that the tendencies of participants remained the same as the previous experiments. We noticed that expressions with higher parameters were classified among the most complex expressions such as expressions 5, 11 and 12.

According to the parameters, these expressions were really the most complicated in this test. In fact, the depth (10) of expression 5 is more significant than the one of expression 12 (6). Also, expression 5 has a very high number of overlapping blocks that affects significantly the complexity of expression 5.

TABLE IX
EXPERIMENT'S RESULTS OF A COLLECTION OF 12 EXPRESSIONS.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	7	6	4	20.56	7.19	2.41	2.75
2	4	3	3	19.88	7.75	3.57	1.66
3	1	4	4	15.25	11.02	0.00	1.37
4	7	9	4	31.75	21.51	5.08	3.63
5	6	11	10	66.38	36.31	16.18	4.57
6	3	1	1	8.75	5.38	0.00	1.05
7	4	2	2	18.44	12.63	4.17	1.58
8	5	7	5	27.04	14.75	5.21	3.24
9	9	5	4	47.69	23.63	9.38	4.71
10	2	10	6	45.88	30.52	6.25	4.19
11	8	5	3	38.15	22.77	6.25	4.06
12	10	16	6	46.13	32.31	13.22	4.89

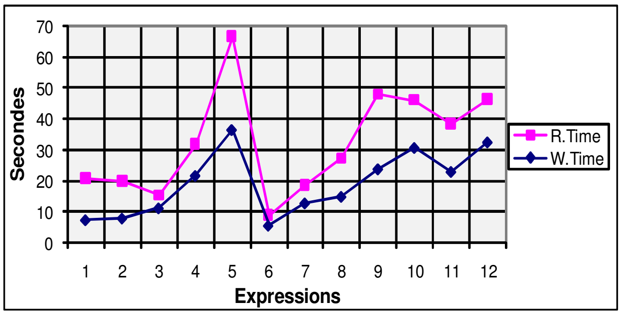


Fig. 12 Curves representing reading and writing times taken by participants during experiment of the 12 expressions.

IX. RESULTS AND APPLICATION

A. *Results Analysis*

According to the results obtained above, our assumption is confirmed that expression complexity depends on the three parameters as well as the mathematical knowledge of participants. We know that the three parameters have a significant impact on the complexity of the expression, but we do not know the weight of each one. Then we try to find these weights by examining graphs and the collected data.

We noted that the three variants follow the same curve. In fact, when the complexity of an expression increased, the reading and writing time increased. However error rate is null when expression is categorized very simple or simple and becomes more significant when expression is more complex. So in this experiment we work on the variant named "subjective complexity".

The principle of this evaluation is to observe complexity of the similar expressions (same number of operands, operators and depth). Table X presents 4 similar expressions that have the same parameters (operands=6, operators=7, depth=4). In this work, we say that these 4 expressions are of model M(6,7,4).

Expressions shown on Table X come from 4 different mathematical branches (i.e. algebra, analysis, statistics & probability and combination). The data of the last column show the average of all complexities estimated by the participants during the evaluations and the data in the last cell is the average of all averages of these complexities.

Since our objective is to find this complexity in a systematic way, it is necessary to find the relation between these parameters (operands, operators, depth).

TABLE X
4 SIMILAR EXPRESSIONS AND ITS AVERAGE COMPLEXITY.

Branch	Operands	Operators	Depth	complexity
Algebra	6	7	4	3.69
Analysis	6	7	4	3.97
Probability	6	7	4	3.52
Combination	6	7	4	3.64
			Average	3.71

We note here that the subjective complexity follows a linear curve then we opt to propose a linear function. Specifically, the idea is to associate a weight for each parameter. The formula that we proposed is:

$$C = x \times P + y \times F + z \times D \qquad (2)$$

where C is the complexity of the expression, P the number of operands, F the number of operators, and D is depth of the tree. x, y and z are the weights of the operands, operators and depth respectively.

Let us consider a mathematical expression which is of model M(6,7,4) then according to Table X, its average complexity is 3.71.

The equation becomes: 3.71=6x+7y+4z

Then, we take a set of similar expressions in order to have more data for determining the values of these unknown variables.

To find the values for these variables, we follow the same method used to collect the data of Table X. The experiments count 72 different expressions which belong to 51 different models. We keep the model that has at least two samples of expressions in our experiments. We found only 13 models which presented in Table XI. Considering these data, we build a problem of 13 equations (see Fig. 13) that illustrates a linear programming problem. It consists of the following three parts:

1) The linear function to be optimized is:
$C = x \times P + y \times F + z \times D$;
2) The problem constraints presented in Fig. 13; 3) the non-negative variables x≥0, y≥0, z≥0.

$$\begin{cases} 2x + y + z \quad = 1.05 \\ 3x + y + z \quad = 1.03 \\ 3x + 2y + 2z = 1.58 \\ 3x + 4y + 3z = 1.99 \\ 4x + 2y + 2z = 1.68 \\ 4x + 3y + 3z = 1.77 \\ 4x + 4y + 4z = 2.26 \\ 4x + 6y + 4z = 2.83 \\ 5x + 4y + 3z = 2.66 \\ 5x + 5y + 3z = 2.96 \\ 6x + 7y + 4z = 3.71 \\ 7x + 6y + 4z = 3.32 \\ 8x + 7y + 3z = 3.83 \end{cases}$$

Fig. 13 Our linear programming problem.

Solving this problem, we get: X=0.39; y=0.10; Z=0.16. Then, the formula becomes:

$$C = 0.39 \times P + 0.10 \times F + 0.16 \times D \qquad (3)$$

In Table XI, we use this formula to validate our approach. Column 5 (average Complexity) presents the results of experiments for several models of expressions. However, column 6 (estimated complexity) shows results of calculation using our formula. The last column shows if two complexities (average and estimated) are equivalent.

Table XII shows the categories of complexity (very simple, simple, average, complex, very complex) as we mentioned in the protocol of evaluation and its corresponding numeric value.

The exhaustive success classification rate (for the 72 available mathematical expressions) is 81.94%. The success classification rate (for the 13 models) is about 84.62%. In fact, expressions of model M(4,4,4) and M(7,6,4) in Table XI are considered as simple and average respectively according to the experiment, but average and complex according to our formula. Then the results of our approach, even if they are not always equivalent to the results of the evaluations, are coherent and acceptable.

TABLE XI
COMPARISON TABLE OF ESTIMATED COMPLEXITY USING OUR FORMULA AND AVERAGE COMPLEXITY ACCORDING TO EXPERIMENTS.

Model	Operands	Operators	Depth	Complexity		Valid hypothesis
				Average	Estimated	
M(2,1,1)	2	1	1	1.05	1.04	YES
M(3,1,1)	3	1	1	1.03	1.43	YES
M(3,2,2)	3	2	2	1.58	1.69	YES
M(3,4,3)	3	4	3	1.99	2.05	YES
M(4,2,2)	4	2	2	1.68	2.08	YES
M(4,3,3)	4	3	3	1.77	2.34	YES
M(4,4,4)	4	4	4	2.26	2.60	NOT
M(4,6,4)	4	6	4	2.83	2.80	YES
M(5,4,3)	5	4	3	2.66	2.83	YES
M(5,5,3)	5	5	3	2.96	2.93	YES
M(6,7,4)	6	7	4	3.71	3.68	YES
M(7,6,4)	7	6	4	3.32	3.97	NOT
M(8,7,3)	8	7	3	3.83	4.30	YES

TABLE XII
THE FIVE CATEGORIES OF COMPLEXITY.

Complexity	Category
C<1.5	Very Simple
1.5 ≤ C<2.5	Simple
2.5 ≤C<3.5	Average
3.5 ≤ C<4.5	Complex
4.5≤C	Very complex

B. *Integration of Expression Complexity in Our System*

The analysis of a mathematical expression is explained as follows. Given a mathematical expression in MathML format for example, 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 operators and syntax tree) then sends parameters to the *Expression Evaluator* to determine its complexity based on equation (3). The parser sends then the expression to the *Expression Encoder* to be translated into its encoded format. This process is shown in Fig. 14.

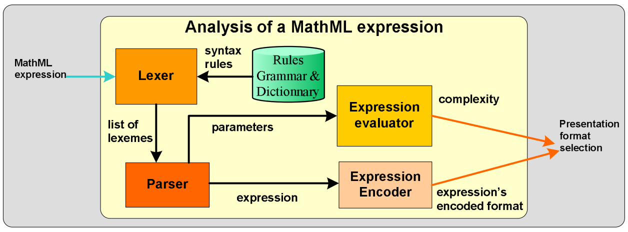
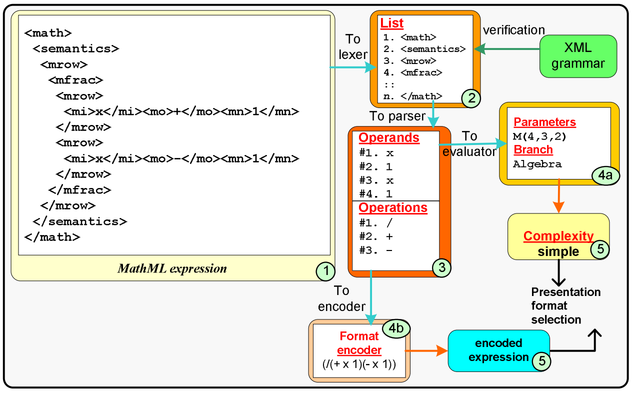


Fig. 14 Structure of the analysis of a MathML expression.

As an example, Fig. 15 shows a sample of fraction (4).

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

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 complexity of the expression (e.g. simple) while the encoder produces the encoded expression. Finally, in step 5, the complexity of the expression is determined and the encoded expression becomes an input to the presentation format selection process.



```
<math>
 <semantics>
  <mrow>
   <mfrac>
    <mrow>
     <mi>x</mi><mo>+</mo><mn>1</mn>
    </mrow>
    <mrow>
     <mi>x</mi><mo>-</mo><mn>1</mn>
    </mrow>
   </mfrac>
  </mrow>
 </semantics>
</math>
```

Fig. 15 A sample analysis of a specimen fraction.

X. 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 current available solutions present mathematical expressions in one presentation format. Also there is no consideration to the contexts 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.

In this work, we have presented the architectural framework of our adaptive system. We provided taxonomy of mathematics and our methodology for determining expression complexity. Also, we have presented the experiment results and the way to integrate complexity in our system.

In our future works, the solution will consider various presentation formats to present the expression in suitable presentation format based on user's situation and expression complexity. Also, future works involve the prototyping of this infrastructure and simulating its performance using several computing systems. Such prototype will also be tested on blind users with various interaction contexts.

APPENDIX

a) Arithmetic

$$3 \times (2 + 4) \tag{1}$$

$$\big((7 \times 6) \times 8\big) \times (3 + 2) \tag{2}$$

$$5 \times (9^2 - 3) \tag{3}$$

$$3 \times (((5 \times) - 2) \div 4) \tag{4}$$

$$(\sqrt{4^3} + 2)^4 \tag{5}$$

$$\frac{2^2}{3^2} \div \frac{4}{5} \times 2 \tag{6}$$

$$\frac{1}{2 + \frac{1}{3}} \tag{7}$$

$$\frac{\sqrt{2} \times \sqrt{5}}{\sqrt{3} \times \sqrt{6}} \times \frac{2}{3} \tag{8}$$

$$\sqrt{\frac{8^3}{11}} - (2^2 - 3^5) \tag{9}$$

$$\sqrt{\sqrt{6}} \tag{10}$$

b) Algebra

$$\sqrt{ab^2c} \tag{I}$$

$$\frac{\sqrt{a \times b + c \times d}}{\sqrt{ab}} \tag{2}$$

$$x^2 + 5mx - m^2 \tag{3}$$

$$(x\sqrt{2} + 1)\sqrt{x + 1} \tag{4}$$

$$x^2 + x \tag{5}$$

$$\frac{1 + x}{1 - x^2} + \frac{1 - x}{1 + x} \tag{6}$$

$$\frac{\sqrt{x} \times \sqrt{y}}{2\sqrt{x + y}} \times \frac{\sqrt{x^2}}{y} \tag{7}$$

$$\frac{\frac{1}{x}}{x + 1} \tag{8}$$

$$\sqrt[3]{x} \tag{9}$$

$$\frac{\sqrt{x}}{x + 1} \times \frac{1}{x} \tag{10}$$

c) Analysis

$$f(x) = \frac{x + 1}{x - 1} \tag{1}$$

$$\frac{2}{\log x} \tag{2}$$

$$\int_0^4 x dx \tag{3}$$

$$y = 3x + 1 \tag{4}$$

$$y = \frac{\sin x}{\sqrt{x}} + \frac{1}{x + 2} \tag{5}$$

$$y = \frac{\sqrt{\sin x}}{2 + x} + \frac{1 + x}{\sqrt{\cos x}} \tag{6}$$

$$y = \frac{x}{2} \times \sqrt{\frac{3}{x}} \tag{7}$$

$$\frac{\log_{10} x + \log_{10} y}{x^2 - y^2} \tag{8}$$

$$\frac{\sqrt{\log(x + 1)}}{\log \sqrt{x^2 - 1}} \tag{9}$$

$$y = (3x + 2)(2x - 1) \tag{10}$$

d) Geometry / Trigonometry

$$sin\, x = \frac{1}{cos\, y} \tag{1}$$

$$cos\, x = \frac{3\sqrt{7}}{7} \tag{2}$$

$$(1 + cos\, 2x)(1 - sin\, 3x) \tag{3}$$

$$(sin\, 2x)^3 + \sqrt{tan\, x} \tag{4}$$

$$|tan\, x| = \sqrt{sin\, x} \tag{5}$$

$$\sqrt{|sin\, x|} \tag{6}$$

$$cos(x + y) - cos\, x\, cos\, y + sin\, x\, cos\, y = 0 \tag{7}$$

$$(cos\, 3x)^2 + 2a\,(sin\, 2x)^2 \tag{8}$$

$$cos\, y = a\, cos(bx + c) \tag{9}$$

$$4\, cos(x - \pi) = cos(3y + \pi) \tag{10}$$

e) Logic

$$\bigwedge_{i=m}^{n} P_i \tag{1}$$

$$\bigwedge_{i=0}^{n} P(a_i) \vee \bigwedge_{j=1}^{m} P(b_j) \tag{2}$$

$$\neg(\neg p \wedge q) \tag{3}$$

$$\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big) \tag{4}$$

$$(\neg B \Rightarrow \neg A) \Rightarrow \big((B \Rightarrow A) \Rightarrow (A \Rightarrow B)\big) \tag{5}$$

$$\neg q \wedge (\neg p \vee q) \tag{6}$$

$$\neg p \vee p = \bigwedge_{x=0}^{\infty} P(x) \tag{7}$$

$$(p \wedge r \vee q) \wedge \neg\big(p \vee (r \wedge s)\big) \vee (p \wedge \neg \wedge s) \tag{8}$$

$$\neg(\neg p \wedge q \vee \neg(r \wedge \neg s) \vee p) \tag{9}$$

$$\neg(p_1 \vee t) = \neg p_2 \vee r \tag{10}$$

f) Statistics & Probability

$$\prod_{i=0}^{m} \sqrt{i} + \frac{k!}{n!\,(k - n)!} \tag{1}$$

$$\frac{1}{n} \times \sum_{k=1}^{n} P(k) \tag{2}$$

$$\sum_{i=1}^{n} i \tag{3}$$

$$\sum_{i=0}^{k} \left(\frac{n_i^2}{np_i} - n_i\right) \tag{4}$$

$$C_n^k \tag{5}$$

$$P(A \cup B) - \big(P(c) + P(d)\big) \tag{6}$$

$$\sum_{k=1}^{n} \sum_{j=0}^{m} P(k)P(j) \tag{7}$$

$$\sum_{i=0}^{n} \prod\nolimits_{j=1}^{m} i \times j \tag{8}$$

$$\sum_{i=0}^{n} x_i \tag{9}$$

$$\left|\sqrt{n + 1}\right| = \frac{\frac{1}{|\mathrm{var}(x) - 2|}}{\frac{2}{n! \times k}} \tag{10}$$

g) Combination

$$2x^3 - 3x^2 + 1 \tag{1}$$

$$x + \log_{10}|x^3| \tag{2}$$

$$\left|\ln(\sqrt{|x|})\right| \tag{3}$$

$$\frac{(1 + x)(\sin x + \cos x)}{x + 3 + \tan x} \tag{4}$$

$$\sqrt{\left|x\left[x + 3(\sqrt{|\ln|x||} - 2x)\right]\right|} \tag{5}$$

International Science Index, Mathematical and Computational Sciences Vol:2, No:3, 2008 waset.org/Publication/11248

$$\int_{0}^{2} x dx \tag{6}$$

$$\sum_{i=1}^{k} n^{i} \tag{7}$$

$$|x| - \frac{\frac{1}{\sqrt{|x|}}}{\frac{5}{x}} \tag{8}$$

$$\prod_{i=1}^{l} i + \sum_{j=1}^{m} \sum_{k=0}^{n} (j \times k) \tag{9}$$

$$\cos\left(\sqrt{|\sin^{-1}(x)|}\right) + \cos\left(\sqrt{|\sin^{-1}(\cos y)|}\right) \tag{10}$$

$$\sum_{i=0}^{l} i + \prod_{j=1}^{m} j + \log_{n}(x) \tag{11}$$

$$\frac{\ln\left|\frac{1}{x}\right| + \sqrt{\left|\frac{x-1}{x+1}\right|}}{\sqrt{|x|}} \times x! \tag{12}$$

REFERENCES

[1] Awde, A., et al., *Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users*, in *2nd International Conference on Grid and Pervasive Computing*. 2007, Lecture Notes in springer-Verlag: Paris, France.

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

[3] Edwards, A.D.N. and R.D. Stevens. *A Multimodal Interface for Blind Mathematics Students*. in *INSERM*. 1994. Paris, France.

[4] Cahill, H., et al., *Ensuring Usability in MATHS*, in *The European Context for Assistive Technology*. 1995, IOS Press: Amsterdam. p. 66-69.

[5] Preddy, M., et al. *Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math*. in *International Conference on Technology and Persons with Disabilities*. 1997. Los Angeles, CA, USA.

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

[7] Ferreira, H. 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*. 2004. Paris, France.

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

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

[10] Schwebel, F. 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*. 2005. Caen, France.

[11] Batusic, M., K. Miesenberger, and B. Stoger. *Labradoor, A Contribution To Making Mathematics Accessible For The Blind*. in *International Conference on Computers Helping People with Special Needs*. 1998. Oldenbourg, Wien, München.

[12] Karshmer, A.I., et al., *Reading and writing mathematics: the MAVIS project*, in *Proceedings of the third international ACM conference on Assistive technologies*. 1998, ACM Press. p. 136-143.

[13] Nemeth, A., *The Nemeth Braille Code for Mathematics and Science Notation*. American Printing House for the Blind, 1972.

[14] Epheser, H., D. Pograniczna, and K. Britz, *Internationale Mathematikschrift fur Blinde, Deutsche Blindenstudienanstalt*. Marburg (Lahn), 1992.

[15] *Commission Evolution du Braille Francais. Notation Mathematique Braille, mise à jour de la notation mathematique de 1971*. 2001.

[16] Garlini, P. and F. Fogarolo, *Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs*, in *Lambda Project*. 2003, University of Padova: Padova, Italy.

[17] Raman, T.V., *Audio System for Technical Readings*. 1st edition ed. Vol. 1410. 1999, Secaucus, NJ, USA: Springer-Verlag New York, Inc. 137.

[18] Rousseau, C., et al., *A Framework for the Intelligent Multimodal Presentation of Information*. Signal Processing Journal, 2006. **86**(12, Special Issue on Multimodal Interface): p. 3696 - 3713.

[19] Demeure, A. and G. Calvary. *Le Modèle d'Evolution en Plasticité des Interfaces: Apport des Graphes Conceptuels*. in *15ème Conférence francophone sur l'Interaction Homme-Machine (IHM 2003)*. 2006. Caen, France: ACM Press.

[20] Awde, A., et al., *Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users*, in *ISIE 2006, IEEE International Symposium on Industrial Electronics*. 2006: Montreal, Canada.

[21] Awde, A., et al., *Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users*, in *2nd International Conference on Grid and Pervasive Computing (GPC 2007)*. 2007, Lecture Notes in springer-Verlag: Paris, France. p. 459-471.

[22] Bellik, Y., *Interfaces multimodales : concepts, modéles et architectures.*, in *LIMSI*. 1995, Université d'Orsay: Paris.

[23] Awde, A., et al., *An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users*, in *Journal of Multimedia (JMM)*. to be published.

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

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

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

[27] Awde, A., et al., *A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users*, in *The First International Conference on Grid and Pervasive Computing*. 2006: Tunghai University, Taichung, Taiwan.

[28] Cormen, T.H., et al., *Introduction to Algorithms*. second ed. 2001: MIT Press and McGraw-Hill.

[29] McCowan, I., et al., *On the Use of Information Retrieval Measures for Speech Recognition Evaluation*, in *IDIAP Reseach Report*. 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. He also has a bachelor's degree in computer science. His research interests include multimodal multimedia for visually-impaired users, machine learning, and mathematics for the blind and human-machine interface.

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.

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.