



# Korean language math-to-speech rules for digital books for people with reading disabilities and their usability evaluation

Joo Hyun Park<sup>1</sup> · Jong-Woo Lee<sup>1</sup> · Jong-Seok Um<sup>2</sup> · Juhye Yook<sup>3</sup> · Kwangki Kim<sup>4</sup> · Soon-Bum Lim<sup>1</sup>

Accepted: 16 November 2020 / Published online: 2 January 2021  
© Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Although Digital Accessible Information System, an international digital talking book standard, is conducting active research on upper level education for people with reading disabilities, special components like images or formulas are difficult to read aloud in Korean. Therefore, this paper defines Korean math-to-speech rules for Contents Mathematical Markup Language based on the formulas used in the Korean middle school and high school freshman curriculum. Based on the proposed rules, a program that turns formulas expressed in Contents Mathematical Markup Language into speech was also designed through Extensible Stylesheet Language Transformations. Finally, a performance test was conducted to determine both the accuracy of the transformed formulas and the clarity of formulas when read aloud.

**Keywords** Reading disability · MathML · Math to speech · Korean math-to-speech rule · Evaluation · Usability

## 1 Introduction

The wide distribution of individual smart devices highlights the importance of accessibility that enables all people to freely use digital devices regardless of any disabilities [1]. In line with such social trends, researchers continue to develop

---

✉ Soon-Bum Lim  
sblim@sookmyung.ac.kr

Joo Hyun Park  
park.joohyun5@gmail.com

<sup>1</sup> Department of IT Engineering, Sookmyung Women's University, Seoul, Korea

<sup>2</sup> Division of Computer Engineering, Hansung University, Seoul, Korea

<sup>3</sup> Department of Rehabilitation Technology, Korea Nazarene University, Cheonan, Korea

<sup>4</sup> Department of IT Convergence, Korea Nazarene University, Cheonan, Korea

accessibility aids such as digital textbooks to improve quality of life for people with disabilities [2].

Digital textbooks and web document contain diverse forms of multimedia contents including text, images, audio, video, and formulas. Currently, most web documents and e-books contain manual descriptions using captions or alttext when generating documents with formulas, so there is a limit to human processing. Also, it has limitations since it is not fully compatible with the Korean language for non-text contents, especially for math formulas. As a result, the formulas in electronic books manufactured in Korea either are omitted or are mentioned by name only, as if they were images. The linguistic structural difference, in particular, creates problems like incorrect word order when reading formulas in Mathematical Markup Language (MathML) into Korean [3].

Also, in Korea, the formal curriculum is changing to e-learning education and learning using digital textbooks. Reference books and supporting materials for learning have also become digital. E-book creator is trying to express in the production process to MathML formulas to maintain accessibility. Elementary and middle school textbooks and reference books in Korea provide about 300,000 math formulas. In order to hear and understand more than about 300,000 math formulas by sound, it is necessary to convert automatically the reading text according to the Hangul word order of expressions expressed in MathML.

In this research, a math-to-speech method of transforming formulas in MathML [4] was developed for audio books for people with reading disabilities. Specifically, the research defines math-to-speech rules capable of expressing mathematical formula for Contents MathML and Extensible Stylesheet Language Transformations (XSLT) [5] instead a plug-in was used for math-to-speech program development in order to enhance the extendibility and speed. The scope of formulas for research was limited to those used in Korean curriculum from seventh to ninth grades. Finally, tests were conducted to verify both the accuracy of the transformed formulas when read in the Korean language and the comprehensibility of math-to-speech formulas when read aloud to textbook users.

## 2 Current practice and research

With DAISY, an international digital audio book standard, formula contents are expressed in MathML [6]. This part of the paper looks into the drawbacks of the application of global math-to-speech services in the Korean language.

### 2.1 Research on math-to-speech conversion

MathPlayer of Design Science Inc., an American company, is a representative math-to-speech service [7]. It renders MathML in web browser format and reads the formula aloud. The program is available on Internet Explorer browser after installing a plug-in. The grammar rule applied to this program is MathSpeak, a math-to-speech style that has been developed through the MathSpeak project by gh-MathSpeak Inc.,

a company established in 2004 [8]. The grammar rule provided by MathSpeak provides three levels of math-to-speech style: verbose, brief, and super-brief, and the user selects the level in accordance with his or her circumstances and demands.

Math Genie is a browser that supports the audio output of the formula designed by Math Genie Inc. [9]. It is a system designed to be used by people with visual disabilities who do not have prior knowledge of braille, and provides all students, including students with visual disabilities, with equal accessibility to educational materials by supporting formula contents encoded in MathML. The synchronization function between audio and graphic rendering enables the Scalable Vector Graphics (SVG)-based reading of formulas and highlights parts of formulas and thus helps math education for students with dyslexia. The program is available in English, French, and Spanish.

WIRIS MathType is an HTML-based JavaScript editor that can write and edit mathematical expressions and display mathematical graphics in Cartesian coordinate systems [10]. WIRIS Math Editor is a browser application that supports both MathML and LaTeX. MathType not only outputs the expression written in the editor through MathML or LaTeX, but also supports output to an image so that you can use the `<img>` HTML tag. In addition, the way to provide access to images is to add appropriate text to the ALT attribute. MathType automatically generates and provides English text for the formula.

MathJax is a cross-browser JavaScript library that displays math notation in a web browser using MathML, LaTeX, and ASCIIMathML markup [11]. MathJax is provided as open-source software under the Apache license. MathJax is downloaded as part of a web page, scans mathematical markup on the page, and enters mathematical information accordingly. So MathJax does not require any software or additional fonts to be installed on the reader system. MathJax can display math using a combination of HTML and CSS or, if available, using the browser's native MathML support. The exact method MathJax uses to typeset math depends on the capabilities of your browser, the fonts, and configuration settings available on your system, and it uses SVG rendering to display formulas on the screen. MathJax supports math accessibility by exposing MathML to assistive technology software and native WAI-ARIA "role" and "text" attributes through APIs.

EyeMath is a cloud-based mobile application that reads mathematical expressions in documents using Amazon AWS' serverless microservices [12]. EyeMath splits the input image into smaller pieces and calls the MathpixOCR API to separate the pieces with only plain text and pieces with mathematical symbols into LaTeX strings. Math-related pieces are added to the Abstract Syntax Tree (AST) and written/parsed in Thai sentences. Plain text fragments are converted to text using Tesseract OCR. Finally, all the pieces were combined to be developed so that the screen reader of the device can read them out loud.

EAR-Math proposed a methodological approach for user-centered evaluation of mathematical rendering against baseline [13]. They defined a set of performance metrics that capture the distance between the reference and the perceived math expressions tree. EAR-Math measured system performance using subdivided error rates based on structural elements, mathematical operators, and numbers and identifiers in equations' syntax trees. This measurement was performed about the Greek audio rendering rules of MathPlayer in a pilot study. This study confirmed that the

structural error rate was higher in mean and variance than the other two metrics and that participants improved second and third when listening to the stimulus.

TechRead's [14] used prosodic to denote nested structures to the synthetic voice. To devised mechanism for mathematical content can be delivered man intuitive manner, participants with sight were asked to choose from a list of prepared answers that matched the audio stimulus. Of the 16 rendered math expressions, three were being trained and the rest were played in three phonetic modes.

## 2.2 Research on non-English math-to-speech transformation

In Japan, Nihon University and the Institute of Systems, Information Technologies, and Nanotechnologies (ISIT) have proposed a new methodology for math-to-speech within DAISY [15]. In this research, some mathematical expressions in Japanese are difficult to express in math-to-speech; these were briefly studied in the course of this research. Ways to address ambiguities in the pronunciation of terminologies when they are used in mathematical and other contexts were investigated.

The three major ambiguities identified are fractional expressions, subscripts or superscripts, and formulas with parentheses. For such examples when the pronunciation could vary according to the context, a methodology using Ruby is proposed to add supplemental information to clarify any ambiguity. Ruby methodology adds phonogram explanations for foreign languages or Chinese characters on printed documents, and can be expressed in either hiragana or katakana.

Japanese researchers have suggested a function called Yomi that can add additional information on formulas with a Ruby tag. Ruby and Yomi are different in terms of their basic concepts. While Ruby is included in the original print, Yomi is not. Therefore, Yomi is not apparent on the DAISY text but can adjust the audio output, and thus, a method using Yomi does not affect the original.

MathReader is a tool that enables the visually impaired people in Thailand to study math or science independently [16]. MathReader automatically reads mathematical symbols or formulas in Thai and consists of four main modules. Converting a mathematical expression in Thai to speech is different from changing plain text due to characters and structures. In the Thai writing system, each term is written in linear form from left to right without space between. Also, consonants, vowels, and tone markers express characters in four levels of depth. On the other hand, mathematical expression is a multi-level system. Superscripts and subscripts can be expanded to multiple levels without restrictions. Therefore, additional words are needed to express a mathematical expression in Thai. The Phrase Identification module divides the input string into normal Thai text and mathematical expression parts. The Thai Text Reader module generates the speech of the Thai text part, while the MathEx Reader module generates the speech of the mathematical expression part. The Math Reader module combines all the separate parts corresponding to the voice of the original input.

Polish reads mathematical expressions in a different way from other languages. In Polish, the expression of a mathematical expression is the same as that of English, but the rules of reading depend on the preference [17]. For example, in a general

formula, the comma serves to separate numbers from numbers, but in Polish, the comma means the decimal separator. Also, the mathematical notation is read differently depending on the situation. Therefore, in this study, a guide to the rules of reading was defined through a survey for teachers who had experience in teaching mathematics formula reading to the visually impaired. Based on the defined reading rules, this study developed a speech output tool considering Presentation MathML structure symbols and mathematical contexts for adjacent node names.

### 2.3 Differences of applying Korean math-to-speech in conventional research

As aforementioned, most of the commercialized math-to-speech service systems are based on DAISY format and provide math-to-speech services in English to people with reading disabilities. When such a system is applied in Korea, its function is disrupted because of the difference of expressing formulas in English and Korean. For example, when expressing a fraction in English, the numerator is read first, while it is read after the denominator in Korea. When such a rule is disregarded, the original meaning is distorted, as in Table 1.

As expressed in Table 1, the formula is read "x over 2 plus 1" in English and "2 bun-eui x plus 1" in Korean, which literally means "x under 2 plus 1." When literally translating the English math-to-speech into Korean, however, the system will read "x bun-eui 2 plus 1" so the user will comprehend it in an entirely different way. In order to prevent such errors, a system with correct rules for each language is essential instead of a literal translation.

In this paper, research on transforming MathML contents into math-to-speech texts was performed. Further, math-to-speech methods were developed for specific target groups including the general public, students, and people with reading disabilities. Since Korean math-to-speech is focused on MathML contents, the web standard, it can be applied not only to electronic books but also to formulas in web documents.

In addition, XSLT was used to design to extract the reading text suitable for Korean word order, focusing on the method of providing a reading service of equations using XSLT in MathPlayer.

In order to extract Korean reading text using the existing MathPlayer method, the following process is performed. First, each single formula created with MathML is converted into a Math Tree, and the tree is expanded to express an expression combined with multiple operators, and the English reading text is extracted by analyzing the tree. The next step is to go through the language translate process and

**Table 1** Example of literal translation of English math-to-speech into Korean

Formula; $\frac{x}{2} + 1$
(English Math-to-Speech) x over 2 plus 1
(Literal translation into Korean) x bun-eui 2 plus 1 (x <small>분의</small> 2 플러스 1)
(Korean Math-to-Speech) 2 bun-eui x plus 1 (2 <small>분의</small> x 플러스 1)

extract the Korean reading text. In this study, the XSLT program was implemented to extract the Korean reading text immediately after generating the extended tree. Korean reading rules were defined, and XSLT templates were defined according to the defined rules. Our system performs math tree generation and expansion tree generation process the same as MathPlayer, but extracts Korean reading text directly from MathPlayer's English reading text extraction step. In the step of extracting the Korean reading text in our system, the xslt written in the Korean reading rule defined in this study is applied (Fig. 1). As XSLT makes a parse tree of the MathML contents, the math-to-speech program was designed without directly coding the parsing process. This has enhanced the convenience of managing and maintaining the math-to-speech rule as the program is designed centering on the math-to-speech rule. In addition, as all frameworks are based on the web standard method, web accessibility has been enhanced.

### 3 Korean math-to-speech rule of formula contents

The final objective of this research is to identify math-to-speech texts for MathML formulas in audio electronic books. The research also has the following detailed targets: first, define phased math-to-speech rules for Contents MathML that can express meanings of mathematical functions; second, transform Contents MathML formula contents to math-to-speech texts using XSLT; and third, design a system that converts the transformed text into Speech Synthesis Markup Language (SSML) [18] content output. Chapter 3 explains the math-to-speech method studied on formulas in Korean math curriculum ranging from seventh to ninth grade.

#### 3.1 Contents MathML node tree analysis

Contents MathML is a method to express formulas for machine-readable and calculation. Figure 2 shows that a single formula is categorized as  $\langle \text{apply} \rangle$ , and the first child node is operator node, followed by operand node.

When there is one operand node, it is right operand node in most cases, but when there are two operand nodes, left is followed by right operand node. When there are more than three operand nodes, they are repeated under a single operator node. For instance, in the case of "a + b + c + d," as an operator of "+" combines the operand nodes,  $\langle \text{plus} \rangle$  becomes the operator node, and the number of operand nodes is four (Fig. 3).

#### 3.2 Defining Korean math-to-speech for Contents MathML formula

According to the formula characteristics, Contents MathML node composition can be categorized into a meta element, operator element, and operand element, and a math-to-speech rule for each has been defined.

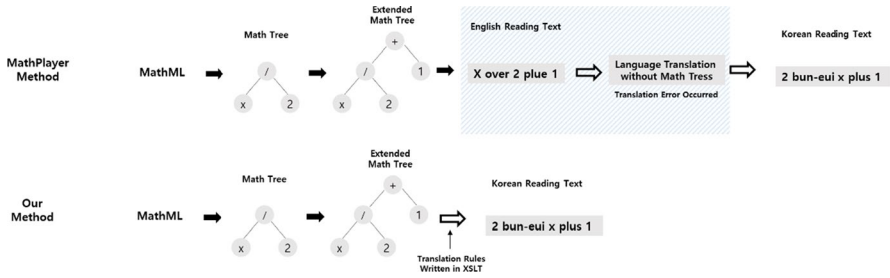


Fig. 1 Differences from MathPlayer

Fig. 2 Contents MathML node tree structure

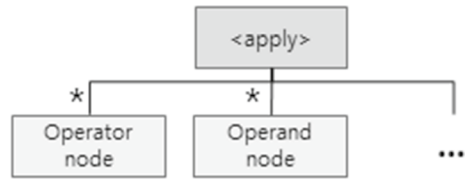


Fig. 3 Contents MathML expressing (a + b + c + d)

```

...
<math>
<apply>
  <plus/>
  <ci> a </ci> <ci> b </ci> <ci> c </ci> <ci> d </ci>
</apply>
...
    
```

3.2.1 Meta element

Meta element consists of <math>, root element, and semantic mapping element that expresses additional formula information.

Contents MathML is expressed as an expression tree that begins with <math>. As <math> is considered a root element that does not have meaning, it is not transformed into Korean math-to-speech.

Semantic mapping element is a node that expresses additional formula information. The nodes that fall under the semantic mapping element are <annotation>, <semantics>, and <annotation-xml>, which are not transformed into Korean math-to-speech as they can be readable through the operation and operand nodes analysis.

3.2.2 Operand element

As an element that expresses the operand of a formula, operand element consists of token element and a non-terminal element. The beginning and ending tags of <math> element determines the scope of the operator. As it uses the prefix notation, the operator is mentioned in the first child element, while the operand is mentioned for the rest of the child elements. The <math> element is not transformed

**Table 2** Math-to-speech rule for <cn> symbols

Contents MathML	Value (symbol)	Korean math-to-speech (English)
<cn> value </cn>	&pi ; &ExponentialE; (or &ee;) &ImaginaryI; (or &ii;) &gamma; &infin; (or &infty;)	파 이 ( Pa-i) 익스포넨셜 e (exponential e) 허수 i (Heo-su i) 감마 (gamma) 무한 대 (Moohandae)

**Table 3** Math-to-speech rule for <ci> type features

Contents MathML	Value	Korean math-to-speech (English)
<ci type = value > a </ci >	Integer Rational Real Complex Constant Function Vector Set	정수 a (Jungsoo a) 유리수 a (Yourisoo a) 실수 a (Silsoo a) 복소수 a (Boksosoo a) 상수 a (Sangsoo a) 함수 a (Hamsoo a) 벡터 a (Bector a) 집합 a (Jibhab a)

**Table 4** Math-to-speech rule for reiterated <apply >

<apply > Reiteration	Korean math-to-speech
<apply > operator argument(s) </apply >	Open parenthesis operator argument(s) Close parenthesis

into Korean math-to-speech. In an operand, either the token element can come as terminal node or formula tree made by <apply > element can come as non-terminal node.

Token factors include numbers, content identifiers, constants, and symbol elements. The first element, numbers, expressed in <cn > produce the same node text value as output. However, the following rules are used when <cn > text value is one of the following (Table 2).

The second variable of <ci > is not transformed into Korean math-to-speech and produces the same value as output. When type feature is used for the actual node, the following words are added to Korean math-to-speech as the type feature can determine variable information (Table 3).

Third, <apply >, a non-terminal node of operand, can be used in reiteration. Excluding the few exceptions mentioned in the next chapter, other formulas can be mapped with the reiteration of <apply > “open parenthesis” and “close parenthesis” in the beginning and end of the formula to reduce the ambiguity of math-to-speech (Table 4).



### 3.2.3 Operator element

As an element expressing the formula operator, operator element appears as the first child factor of  $\langle \text{apply} \rangle$ . The math-to-speech rule for operator element has been defined targeting the formula operator in math curriculum for seventh–ninth grade. Operator was largely categorized into basic element, arithmetic, algebra and logic, relations, set theories, and elementary classical functions. The math-to-speech rule for each is shown in the following table. Among the expressions of Table 5, a and b refer to terminal nodes, while [a] and [b] refer to non-terminal nodes. The terminal nodes are pronounced as they are, but non-terminal nodes are applied with the aforementioned rule of adding “open parenthesis” and “close parenthesis.”

### 3.2.4 Special elements of Korean math-to-speech

The aforementioned math-to-speech rules need additional processing including propositions and elements. First, in Korean, the propositions change into various forms (“은/는, 이/가, 과/와, 을/를”) depending on the final consonant of the previous syllable. Therefore, the proposition should be pronounced differently according to the final consonant of the previous syllable. Second, in the Contents MathML, element refers to “minus sign” when there is one operand, and when there are two operands, it refers to the difference of each operand. Therefore, when there is one operand, “minus” should go in front of the operand, and when there are two operands, “minus” should be in between the operands in the oral speech. Third, when there are duplications of more than two operators, the repeating rule is applied in the Korean math-to-speech. For instance, as for element “[a] plus [b] (plus [n])\*,” “plus [n]” is repeated. The operators that need to be repeated include plus and multiplication operators, greatest and least value operators, greatest common divider and least common multiple operators, and operator elements of union set, intersection, relational algebra, and statistics. Lastly, there is the rule of grouping pronunciation. When reiteration  $\langle \text{apply} \rangle$  factor is used among the operand element, unclear distinction of child and parent factors can lead to confusion. The grouping rule refers to adding the modifier of “open parenthesis” and “close parenthesis” in the front and back in order to restrict the scope of the formula.

To avoid duplicative interpretation of the math-to-speech of  $\langle \text{apply} \rangle$  for child factors, grouping rule must be applied to certain stages. However, for elements that have their official names including fraction, root, and log, a different grouping rule must be applied because the same rule can cause confusion in identifying the overall formula structure. In this case, general users can easily be reminded of the meaning when they hear official names like denominator, numerator, square root, and base, grouping rules have been defined as given in Table 6.

**Table 5** Korean math-to-speech rules for operator

Basic element	Contents MathML expression	Mathematical symbols	Korean math-to-speech (English)
Arithmetic, algebra, and logic	$\langle fn \rangle [f] \langle /fn \rangle [x] \langle inverse \rangle$ $\langle divide \rangle a b$ $\langle divide \rangle [a] [b]$ $\langle max \rangle [a] [b] [n]^*$ $\langle min \rangle [a] [b] [n]^*$ $\langle minus \rangle [a]$ $\langle minus \rangle [a] [b]$ $\langle plus \rangle [a] [b] [n]^*$ $\langle power \rangle a b$ $\langle power \rangle [a] [b]$ $\langle rem \rangle [a] [b]$ $\langle times \rangle [a] [b] [n]^*$ $\langle root \rangle [b]$ $\langle root \rangle \langle degree \rangle$ $[a] \langle degree \rangle [b]$ $\langle gcd \rangle [a] [b] [n]^*$	$f(x)$ $A^{-1}$ $\frac{a}{b}$ $\max \{a, b, n^*\}$ $\min \{a, b, n^*\}$ $-a$ $a-b$ $a+b(+n)^*$ $a^b$ $a \text{ mod } b$ $ab(n)^*$ $\sqrt{b}$ $\sqrt[n]{b}$ $\text{gcd}(a, b, n^*)$	함수 f 괄호 열고 [x] 괄호 닫고 (function f, open parenthesis [x] close parenthesis 역 (inverse) 분수 시작 b분의 a 분수 끝 (fraction start b under a fraction end) 분수 시작 분모 [b] 분모 끝 분의 분자 [a] 분자 끝 (fraction start denominator [b] denominator end, under numerator [a] numerator end) [a]와/과 [b](와/과 [n]) <sup>*</sup> 의 최댓값 (a) and [b] (and [n]) <sup>*</sup> maximum value) [a]와/과 [b](와/과 [n]) <sup>*</sup> 의 최솟값 (a) and [b] (and [n]) <sup>*</sup> minimum value) 마이너스 [a] (minus [a]) [a] 마이너스 [b] ([a] minus [b]) [a] 플러스 [b] (플러스 [n]) <sup>*</sup> ([a] plus [b] (plus [n]) <sup>*</sup> ) a의 b승 (A to be power of b) [a]의 지수 [b] 지수 끝 승 ([a] quotient [b] quotient end power) [a]을/를 [b]로 나눈 나머지 ([a] divided by [b] and the remainder) [a] 곱하기 [b] (곱하기 [n]) <sup>*</sup> ([a] times [b] (times [n]) <sup>*</sup> ) 루트 시작 [b] 루트 끝 (root begin [b] root end) 루트 시작 [b]의 [a] 제곱근 루트 끝 (root begin [b]'s [a] root, root end) [a]와/과 [b] (와/과 [n]) <sup>*</sup> 의 최대공약수 ([a] and [b] (and [n]) <sup>*</sup> 's greatest common divider)

Table 5 (continued)

Contents MathML expression	Mathematical symbols	Korean math-to-speech (English)
<abs/> [a]	a	절대값 시작 [a] 절대값 끝
<lcm/> [a] [b]	lcm (a, b, n)*	(absolute value begin [a] absolute value end) [a]와/과 [b] (와/과 [n])*의 최소공배수
<factorial/> [A]	A!	[A] 팩토리얼 ([A] factorial)
<implies/> [P] [Q]	p⇒q	[P] 이면 [Q]이다 (if [P] then [Q])
<eq/> [a] [b] [n]*	a=b=...=n	[a]은/는 [b](=)고 [n]*이다 ([a] is [b](and is [n])* )
<neq/> [a] [b]	a≠b	[a]은/는 [b]아니다 ([a] is not [b])
<gt/> [a] [b] [n]*	a>b>...>n	[a]은/는 [b]보다 (크고 [n-1]은/는 [n]보다)* 크다 ([a] is than [b] (greater [n-1] is than [n])* greater)
<lt/> [a] [b] [n]*	a<b<...<n	[a]은/는 [b]보다 (작고 [n-1]은/는 [n]보다)* 작다 ([a] is than [b] (less [n-1] is than [n])* less)
<geq/> [a] [b] [n]*	a≥b≥...≥n)*	[a]은/는 [b]보다 (크거나 같고 [n-1]은/는 [n]보다)* 크거나 같다 ([a] is than [b] (greater or equal [n-1] is than [n])* greater than equal)
<leq/> [a] [b] [n]*	a≤b(≤n)*	[a]은/는 [b]보다 (작거나 같고 [n-1]은/는 [n]보다)* 작거나 같다 ([a] is than [b] (less or equal [n-1] is than [n])* less or equal)
<equivalent/> [a] [b]	a≡b	[a]와/과 [b]은/는 항등이다 ([a] and [b] is equivalent)
<approx./> [A] [B]	A≈B	[A]은/는 [B]와 근사하다 ([A] and [B] are approximate)
<set/> [a] [b] [n]*	set {a, b, ..., n}	집합 [a] [b] ([n])* (Set [a] [b] ([n])* )
<union/> [A] [B] [n]*	A∪B∪...∪n	[A] 합집합 [B] (합집합 [n])* ([A] union [B] (union [n])* )
<intersect/> [A] [B] [n]*	A∩B∩...∩n	{A} 교집합 [B] (교집합 [n])* ([A] intersection [B] (intersection [n])* )
<in/> [a] [A]	a ∈ A	[a] 은/는 [A]의 원소이다 ([a] is [A]'s element)
<notin/> [a] [A]	a ∉ A	[a] 은/는 [A]의 원소가 아니다 ([a] is not [A]'s element)
<prsubset/> [A] [B]	A ⊂ B	[A]은/는 [B]의 부분 집합이다 ([A] is [B]'s subset)
Theory of sets		

Table 5 (continued)

Elementary classical functions	Contents MathML expression	Mathematical symbols	Korean math-to-speech (English)
	$\langle \text{notprsubset} \rangle$ [A] [B]	$A \not\subseteq B$	[A]은/는 [B]의 부분집합이 아니다 ([A] is not [B]'s subset)
	$\langle \text{sin} \rangle$ [x]	$\sin x$	사인 [x] (sine [x])
	$\langle \text{cos} \rangle$ [x]	$\cos x$	코사인 [x] (cosine [x])
	$\langle \text{tan} \rangle$ [x]	$\tan x$	탄젠트 [x] (tangent [x])
	$\langle \log \rangle \langle \log \text{base} \rangle A \langle / \log \text{base} \rangle B$	$\log_A B$	로그 A의 B 로그 끝 (Log A's B log end)
	$\langle \log \rangle \langle \log \text{base} \rangle [A] \langle / \log \text{base} \rangle [B]$		로그 밑 시작 [A] 밑 끝의 [B] 로그 끝 (Log base begin [A] base end's [B] log end)

Bold represents the text of the title character

## 4 XLST transform program of Contents MathML into Korean math-to-speech

### 4.1 Math-to-speech transformation process

The math-to-speech transformation process of the formula contents is shown in Fig. 4. The mathematical expression in Contents MathML is read through XSLT, which then transforms it into text form. After this process, the text form is processed into SSML file speech output (Figs. 5 and 6).

### 4.2 XSLT design (Fig. 5)

XSLT designed a math-to-speech program for Contents MathML based on the aforementioned math-to-speech conversion rules. The Contents MathML's Korean Math-to-Speech Rules Library was built based on the reading rules defined in Chapter 3. In this library, Korean readings text corresponding to each node according to the rules is defined to be converted using `xsl: template`. The formulas expressed in Contents MathML are read through XSLT and transform them into text form via the transformation template. Then, the text is processed into SSML file speech output. Through this process, Contents MathML expressions can be converted into audio format.

#### 4.2.1 Top level node transformation of node tree

Contents MathML expressions begin with a root element of `<math>`. When the formula is a single element, meaning it is a number, variable, or constant, it begins with `<cn>`. The template is applied with the math-to-speech rule for processing terminal nodes. When the formula is not a single element, it begins with an element of `<apply>`. Therefore, the node tree normalization module begins with template mapping for `<apply>`. Operator element comes as the first child of `<apply>` node, and operand node comes as the brother node of the operator node. Such node tree sequences must be considered when mapping the math-to-speech rule. Each operator template is designed by applying the math-to-speech rules defined in the previous chapter, and in the node tree normalization module, the appropriate template for the operator, first child of `<apply>`, is used for mapping.

#### 4.2.2 Transformation for general math-to-speech rule processing

In Contents MathML, the operator node can be divided into two large features depending on its brother node or child node. To apply the math-to-speech rule, the child node must first be identified followed by the application of the appropriate math-to-speech template for child node, and then the brother node needs to be identified. When processing the brother node, the math-to-speech template is determined based on the number of the brother nodes. When it is a single node without brother nodes, it is a terminal node among the operand node, so the appropriate template for

**Table 6** Grouping rules

Content MathML	Mathematical symbols	Korean math-to-speech (English)
<apply > .. <apply > operator argument	-	괄호 열고 operator argument 괄호 닫고 (Open parenthesis, operator argument, close parenthesis)
</apply > .. </apply > <divide/> [a][b]	$\frac{a}{b}$	분수 시작 분모 [b] 분모 끝 분의 분자 [a] 분자 끝 (Fraction begin denominator [b] denominator end, under numerator [a] numerator end)
<root/> <degree > [a] </degree > [b]	$\sqrt[b]{a}$	루트 [b]의 [a] 제곱근 루트 끝 (Root [b]'s [a] square root, root end)
<log/> <logbase > [A] </logbase > [B]	$\log_a b$	로그 밑 시작 [A] 밑 끝의 [B] 로그 끝 (Log base begin [A] base end's [B] log end)

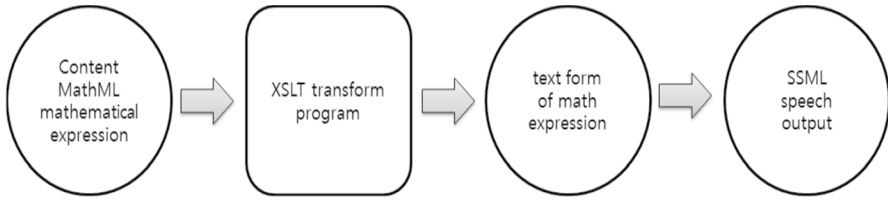


Fig. 4 Math-to-speech program module

Fig. 5 Top level node transformation—applying template to <apply> node

```

<xsl:template match="apply">
  <xsl:apply-templates select="child::*[1]">
<xsl:choose>
  <xsl:when test=".[name()='abs']">
    <xsl:apply-templates select="."/>
  </xsl:when>
  ...

```

```

<xsl:template match="approx">
  <xsl:for-each select="following-sibling::*">
    <xsl:if test=".[name()='ci']">
      <xsl:apply-templates select="."/>
    </xsl:if>
    <xsl:if test=".[name()='cn']">
      <xsl:apply-templates select="."/>
    </xsl:if>
    <xsl:if test=".[name()='apply']">
      <xsl:apply-templates select="."/>
    </xsl:if>
    <xsl:if test="position()=last()"> 은/는 (is/is) </xsl:if>
    <xsl:if test="position()=last()"> 와/과 근사하다(is approximate with)
    </xsl:if>
  </xsl:for-each>
</xsl:template>

```

Fig. 6 Template for processing general math-to-speech rule

the terminal node is used for mapping. When there is one brother node, it is a node that has a left operand, so the operator node must first be mapped, followed by the mapping of the operand node template. When there are two brother nodes, meaning there are both left and right operand nodes of the operator node, the calculation sequence is first determined through the Korean math-to-speech rule and applies the appropriate template for the node (Fig. 6).

In Contents MathML, there are grouping and repetition rules as special rules. A representative example of grouping rule is fraction. As shown in the fraction template in Fig. 7, “fraction begin” and “fraction end” come in the very beginning and end of the fraction, respectively, and if the numerator or denominator is terminal node, then the node value is added to the location of the speech output. When the numerator or denominator is non-terminal node, “denominator”/“numerator,” and

**Fig. 7** Example of template for processing special math-to-speech rule—grouping rule

```

</xsl:if>
...
<xsl:if test=".[name()='apply']">
    분모 시작 (fraction begin)
    <xsl:apply-templates select="."/ >
    분모 끝 (fraction end)
</xsl:if>
...

<xsl:template match="times">
    <xsl:for-each select="following-sibling::*">
        <xsl:if test=".[name()='ci']">
            <xsl:apply-templates select="."/ >
        </xsl:if>
        ...
        <!-- repating read _ check last node -->
        <xsl:if test="position() != last()">
            곱하기(multiplication)
        </xsl:if>
    </xsl:for-each>

</xsl:template>

```

**Fig. 8** Example of template for processing special math-to-speech rule—repetition rule

“denominator end”/“numerator end” are added to accurately define the scope of the denominator and numerator.

When there are more than three brother nodes, repetition rule is applied. As shown in Fig. 8, the template for general repetition math-to-speech rule is converted consecutively for each operand, and the operator in between the operands is added to the speech output. When the four fundamental arithmetic operations are repeated, for instance “ $a*b*c*d$ ,” only the operator is repeated, for example, “a times b times c times d.” However, when there is a sign of inequity, for instance “ $a > b > c$ ,” the speech output will produce “a is larger than b, larger than c” which will make the meaning ambiguous. Therefore, in order to eradicate such ambiguity, the repetition of the operand is used.

For Korean math-to-speech, there is a special rule for propositions of operands. This is determined by the final consonant of the operand node. When there is a final consonant, certain propositions (“은,” “이,” and “을”) are used, while other



```

<xsl:template match="/" /math">
<!-- SSML output -->
<speak version="1.1" xmlns="http://www.w3.org/2001/10/synthesis"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
  http://www.w3.org/TR/speech-synthesis11.xsd"
  xml:lang="ko">
  <p>
  <xsl"apply-templates select />
  </p>
</speak>
</xsl:template>

```

**Fig. 9** Template for audio style application

propositions (“는,” “가,” and “를”) are used when there is no final consonant. In addition, the grouping rule is added according to the complexity of the operand, and the rule is varied according to the type of the operator.

### 4.2.3 Application of audio style

To apply the audio style module, the system enables the speech output in SSML file, which is most widely used. When applying XSLT, SSML header and tag are combined with the final speech output to produce SSML file output (Fig. 9).

## 4.3 XSLT design and sample page

In this study, a sample page was produced to confirm the correct application of Korean reading rules. The sample page generated Contents MathML to read the formulas, and created the presentation mathml of the formulas written in Contents MathML to render the math formulas on the screen so that the math formula were displayed on the web browser screen. In this study, a sample page was produced to confirm the correct application of Korean reading rules. Currently, Contents MathML does not support rendering images in browsers. Therefore, the sample page created Contents MathML and Presentation MathML of the equation to be read, and the image of the equation is displayed on the web browser screen. The XSLT program implemented in this study transformed Contents MathML and extracted the text of the reading. The expression expressed as Contents MathML is converted to the reading text by calling the translation template of the operator using the XSLT program implemented in this study. The extracted text was used to output an SSML file for voice support. Figure 10 is a screen of a Korean math-to-speech text. Clicking on the “read” button, the user can listen to the speech output.

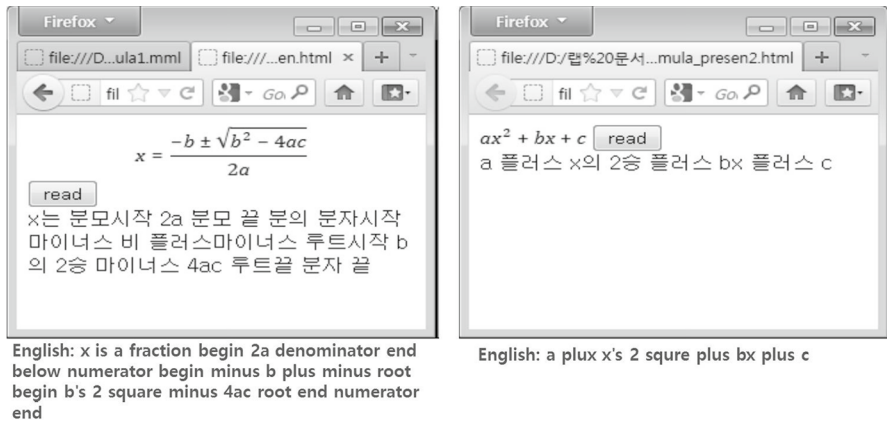


Fig. 10 Screen of Korean math-to-speech text

## 5 Usability evaluation

### 5.1 Usability evaluation plan

In this research, the Korean math-to-speech rule for mathematical contents expressed in Contents MathML within electronic books and web contents targeted for people with disabilities was studied in order to enhance the accessibility of people with reading disabilities to mathematical contents, and a math-to-speech program was designed. Therefore, it is necessary to verify how the input formula is accurately converted to Korean text and how well the defined reading rules are understood by the user.

In this Chapter, accuracy test was carried out to see how accurately the formula has been converted, and a comprehensibility test was carried out to see how well users understand the speech output.

Five participants participated in usability evaluation according to the most common usability methodology, a laboratory test requiring 4–10 relatively few participants [19]. All users who participated in the experiment are 20s college students. And that the results of the experiment were used in the paper and signed the agreement. The experimenter signed a consent form agreeing to use age, occupation, experimental content, and experimental results in the paper. Informed consent was obtained from the participants involved in the study.

The Accuracy Test evaluates the accuracy of the formulas converted to text for 15 mathematical formulas presented in the Korean middle and high school mathematics curriculum. At this time, after making 15 formulas in MathML, convert them to math-to-speech program to check the execution result. Comprehensibility test evaluates the accuracy by listening and reading dictation on each of the five formula examples according to the difficulty level upper, middle, and lower. The scoring standard for dictation is measured as correct only when 100% of the formula read is written.

Recently, researches are underway to convey mathematical expressions by voice for users with visual impairments. Among them, related research EAR-Math[X] proposed a methodological approach for user-centered evaluation of mathematical rendering against baseline. EAR-Math measured system performance using subdivided error rates based on structural elements, mathematical operators, and numbers and identifiers in equations tree. We set the performance metric by referring to the performance metric of EAR-Math. Our study measured accuracy based on language structure translation, operators translation, and grouping rule based on the analysis that the problem occurring in language translation is the biggest problem in understanding mathematical expressions.

## 5.2 Accuracy test

This experiment aims to check how accurately the proposed system extracted Korean reading text compared to MathPlayer. The evaluation method is as follows. The experiment measures on three metrics. The first metric is accuracy of language structure translation. After completing a given task, the number of correctly translated language structures is scored for the total number of leading text language structures in the equation. The second metric is operator translation accuracy, and the measurement method scores the number of single correct operator translations for the total number of single operators. The last metrics is grouping accuracy which is measured the number of groupings applied correctly to the total number of grouping rules. The accuracy score is calculated by averaging the values of the three items.

$$\begin{aligned}
 \text{Language structure translation} &= \frac{\# \text{Correct structure}}{\# \text{Total structure}} \times 100 \\
 \text{Operator translation} &= \frac{\# \text{Correct operator translation}}{\# \text{Total operator}} \times 100 \\
 \text{Grouping rules} &= \frac{\# \text{Grouping applied correctly}}{\# \text{Total grouping rules}} \times 100 \quad . \quad (1) \\
 \text{Accuracy} &= \text{Average}(\text{Language structure translation} \\
 &\quad + \text{Operator translation} + \text{Grouping rules})
 \end{aligned}$$

The tasks used for evaluation are as follows.

First, we convert 15 equations to MathML. After extracting the English reading text by playing the converted formula in MathPlayer, we calculate the result of translating the extracted English text into Korean and calculate the accuracy score for the extracted Korean reading text through our system. All of the mathematical formula examples of the Korean middle school and high school mathematics curriculum have been converted using the proposed Korean math-to-speech method. Table 7 shows the formula used in the task and the correct reading text for the formula, and the score is calculated by comparing it with the table.

After conducting an experiment for a total of 15 equations, an average was calculated for each measurement metric. Table 8 shows the results of the accuracy test of

**Table 7** Example for Each Category

Level	Representative formula example	Korean math-to-speech (English)
Low	$0.7 + 1.6$ $3a < b < \frac{c}{2}$ $A \cup B = B \cup A$ $\frac{a+b}{c+3} = 20$	영점 칠 플러스 일점 육 Zero point seven plus one point six 3a는 b보다 작고 b는 2분의 c 보다 작다 (3a is less than b, b is less than 2 under c) A 합집합 B는 B 합집합 A (A union B is B union A) 괄호 열고 분모 시작 c 플러스 3 분모 끝의 분자 시작 a 플러스 b 분자 끝 괄호 닫고는 20 (open parenthesis denominator start c plus 3 denominator end under numerator start a plus b numerator end close parenthesis is 20)
Middle	$ a  -  b $ $\sqrt{a + b + 2}$ $ab(n)^*$ $A^{-1}$ $\log_a \neq 1$ $a \notin A$	절대값 시작 a 절대값 끝 마이너스 절대값 시작 b 절대값 끝 Absolute value start a absolute value end minus absolute value start b absolute value end 루트 시작 a 플러스 b 플러스 2 루트 끝 Root start a plus b plus 2 root end a 곱하기 b 괄호 열고 n 괄호 닫고 제곱 a times b times open parenthesis n close parenthesis's star 대문자 에이의 역 Reverse of the capital A 괄호 열고 로그 시작 a 로그 끝은 1이 아니다 Open parentheses and start log a log end is not 1 A는 A의 원소가 아니다 A is not an element of A

Table 7 (continued)

Level	Representative formula example	Korean math-to-speech (English)
High	$b = c \sin B$	b는 c 곱하기 사인 B b is c times sine B
	$f(x) = 3x + 4$	함수 f 곱호 열고 x 곱호 닫고 는 3 곱하기 x 플러스 4 Function f open parenthesis x close parenthesis is 3 times x plus 4
	$f \circ g$	f 합성 g f composite g
	$\cot \frac{\pi}{6} + 1$	곱호 열고 코탄젠트 곱호 열고 분수 시작 6분의 파이 분수 곱 곱호 닫고 곱호 닫고 플러스 1 Open parenthesis cotangent open parenthesis fraction start pi under 6 end fraction close parenthesis plus 1
	$\log_{(a+b)}(x+y)$	로그 밑 시작 a 플러스 b 밑 끝의 x 플러스 y 로 곱 곱 Log Begin a plus b End of x plus y Log End

**Table 8** Accuracy task result and type of error of MathPlayer test

Level	Representative formula example	Korean math-to-speech (English, literal translation to Korean)	Result
Low	$0.7 + 1.6$	0.7 plus 1.6 0.7 더하기 1.6	C
	$3a < b < \frac{c}{2}$	3 a less than b less than c over 2 3a는 b보다 작고 c분의 2보다 작다	LT
	$A \cup B = B \cup A$	A union B equals B union A A 합집합 B는 B 합집합 A와 같다	C
	$\frac{a+b}{c+3} = 20$	fraction numerator A plus b over denominator C plus 3 end fraction equals 20 분수 분자 a 더하기 b 분의 분모 C 더하기 3 끝 분수는 20과 같습니다	LT, GL
	$ a  -  b $	open vertical bar a close vertical bar minus open vertical bar b close vertical bar 열기 수직 막대 a 닫기 수직 막대 마이너스 열기 수직 막대 b 닫기 수직 막대	OT
Middle	$\sqrt{a + b + 2}$	square root of a plus b plus 2 end root a 더하기 b 더하기 2 끝 근의 제곱근	OT
	$ab(n)^*$	ab left parenthesis n right parenthesis asterisk times ab 왼쪽 괄호 n 오른쪽 괄호 별표 시간	LT, OT
	$A^{-1}$	A to the power of negative 1 end exponent A의 끝 지수가 음수 1 제공	LT, OT
	$\log_a \neq 1$	log subscript a not equal to 1 로그 첨자 a가 1과 같지 않음	GL
	$a \notin A$	a not an element of A a는 A의 요소가 아님	C
High	$b = c \sin B$	b equals c sin B b는 c sin B와 같습니다	OT
	$f(x) = 3x + 4$	f open parentheses x close parentheses equals 3 times plus 4 f 여는 괄호 x 닫는 괄호는 3 곱 더하기 4와 같습니다	OT
	$f \circ g$	f ring operator g f 링 연산자 g	OT
	$\cot \frac{\pi}{6} + 1$	cot open parentheses pi over 6 close parentheses plus 1 코탄젠트 여는 괄호 파이분의 6 닫는 괄호 플러스 1	LT
	$\log_{(a+b)}(x + y)$	log subscript a plus b end subscript open parentheses x plus y close parentheses 로그 아래첨자 a 플러스 b 끝 아래첨자 는 괄호 x 플러스 y 닫는 괄호	LT, GL

MathPlayer. In the result cell, C means correct, LT means language translation error, OT means operator translation error, and GL means grouping rule error. In Korean math-to-speech cells, the underline indicates the part of the error.

**Table 9** Accuracy task result

Metrics	MathPlayer	Our system
Language structural translation	77.7	100
Operator translation	68.3	100
Grouping rules	88.9	100
Overall accuracy	76.6	100

**Table 10** Test method and measurement method

Test method	<ol style="list-style-type: none"> <li>(1) Participants are asked to dictate a total of 15 formulas, five from each level of difficulty of high, medium and low</li> <li>(2) Participants are asked to dictate formula read with a 30% faster speed than normal book reading speed</li> <li>(3) Three days after the 1st test, the formulas are read again with the normal book reading speed</li> <li>(4) Three days after the 2nd test, the formulas are read again with a 30% slower speed than normal book reading speed</li> </ol> <p>* For each test, the formulas are read only once</p>
Measurement method	<ol style="list-style-type: none"> <li>(1) Dictated formulas are compared with the original to calculate the number of correct dictation. The correct answer is calculated only when comparing each constant, operator and etc. If one is wrong in each constant, operator, etc., it is scored as an incorrect answer</li> <li>(2) Answers are averaged according to the level of difficulty</li> </ol> <p>Comprehensibility= <math>Average(\frac{\#Correct\ Dictation}{\#Total\ Dictation})</math></p>

MathPlayer scored 77.7, 68.3, and 83.9 points in language structural translation, operator translation, and grouping rules accuracy measurements, resulting in a total average of 90.1 points. On the other hand, our system scored 100 points for all the metrics measurements (Table 9). As a result of this experiment, it was confirmed that our system not only reduced the number of work steps compared to MathPlayer, but also accurately output in Korean.

### 5.3 Comprehensibility test

A comprehensibility test was carried out to measure the level of users' understanding of the speech output produced by the suggested Korean math-to-speech program. To measure comprehension metrics, we conducted a dictation test. Five users were asked to dictate the formula read by the system as they were read with different speed each time with a three-day interval. Difficulty of math was divided into low, middle, and high according to the curriculum and complexity of the formula. Five questions were asked according to the level of difficulty, and the answers were scored based on the completeness of the dictation. Experimental methods and measurement methods are as follows, and the formulas used in the experiment are shown in Table 8 as same with accuracy test task (Table 10).

**Table 11** Formula dictation test results

Condition 1	30% faster than normal book reading speed (reading speed of electronic viewer)					
	User 1	User 2	User 3	User 4	User 5	Average (percentage score)
Low	4	4	4	4	4	4 (80)
Medium	1	0	0	1	0	0.4 (8)
High	1	0	0	0	0	0.2 (4)
Condition 2	Normal book reading speed (normal)					
Low	4	4	4	4	4	4 (80)
Medium	1	0	1	2	1	1 (20)
High	1	1	1	0	1	0.8 (16)
Condition 3	30% slower than normal book reading speed (slow)					
Low	4	4	4	4	4	4 (80)
Medium	2	1	2	3	2	2 (40)
High	1	1	1	2	2	1.4 (28)

**Table 12** Number by error type according to the difficulty of the formula

Level/error type	A	B	C
Low	0	1	14
Medium	0	10	48
High	10	16	37

The similar formulas were read with three different speed rates: 30% faster than normal book reading speed, normal book reading speed, and 30% slower than normal book reading speed. The participants were asked to dictate the formula and their answers were measured to find out the accuracy, which was finally used to determine their level of comprehension. The average score for each question was calculated by categorizing the questions into three levels of difficulty: high, medium, and low. The results of the test are shown in Table 11.

The above results show that the more complex the formula, the more difficult for users to comprehend, and thus the less accurate the dictation answers. Also, when the speed of reading is faster, the comprehensive level is lower, and the average number of correct dictation answers is also lower.

Looking at the types of errors, there were three major types of errors: a type that was not fully dictated (A), a type that left partial blanks (B), and a type (C) that dictated symbols or numbers in some wrong places. Table 12 shows the number of error types for each difficulty of the formula.

The averages for each reading speed and formula level of difficulty have been produced in a graph shown in Fig. 11. Results showed that for low level of difficulty, the users were able to comprehend all of the formula regardless of the speed. However, when the difficulty level was raised above medium, the comprehension level was apparently lower in accordance with the faster speed. It can be deduced



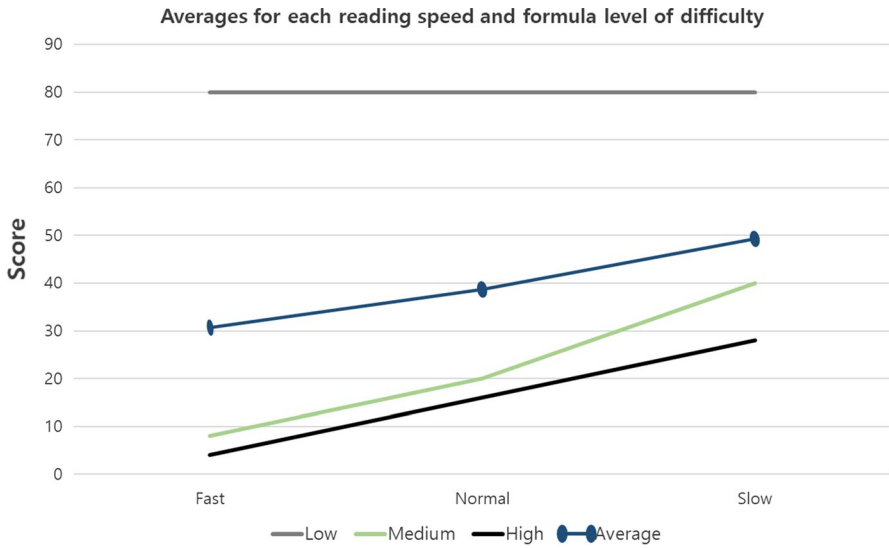


Fig. 11 Average graph according to formula level of difficulty and reading speed

that very simple formulas can be easily comprehended even at fast speeds, but as the level of difficulty increases, the formula must be read at a slower pace. Based on the suggestions of the test participants, further research seems necessary for reading methods including reading by punctuating.

### 6 Conclusion

The electronic books that are currently manufactured in Korea can only convert the texts and thus provide limited audio services for people with reading disabilities as the mathematic formula and symbols are not converted. Internationally, however, there is much research on math-to-speech services in order to grant enhanced mathematical accessibility to people with reading disabilities. Such research must be carried out by each nation as each language has its own ways of expressing mathematical formulas.

In this research, the Korean math-to-speech rule for mathematical contents expressed in Contents MathML within electronic books and web contents targeted for people with disabilities was studied in order to enhance the accessibility of people with reading disabilities to mathematical contents, and a math-to-speech program was designed. Formulas included in the Korean math curriculum from middle school to freshman of high school were defined, and XSLT was used to design a math-to-speech converting system, and it was confirmed that the program could accurately convert the formulas by providing SSML contents with the addition of tags onto the transformed math-to-speech text as the final output. In addition, the accuracy test confirmed that the contents were 100% accurately converted according to the rules. Through this test, it was confirmed that the mathematical expressions

were accurately represented according to the Korean word order, compared to the reading program of the equation used in the existing English-speaking languages. The comprehensibility test also confirmed that the speed of reading had an impact on the level of comprehension. As a result, this paper's proposed approach is a valuable contribution to the field of providing access to the population of blind mathematics.

This study has the following contributions. First, it maximized web accessibility by defining the entire program framework ranging from input, transformation to output as web standard. This enables the math-to-speech output by using web document formula. Second, Korean math-to-speech rule for Contents MathML was defined to enhance the accessibility for people with reading disabilities to mathematical contents expressed on the web. Going forward, defining the math-to-speech rule for mathematical formulas expressed in MathML, and rules for each circumstance must be established. In addition, according to the degree of mathematics learning of the visually impaired students, the comprehension test for each level should be conducted and the definition of reading rules for each level should be performed.

**Acknowledgements** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1A4A1025559).

## References

1. Lee KH, Lee TE, Lee JW, Lim SB (2013) A design of mobile e-book viewer interface for the reading disabled people. *J Korea Multimed Soc* 16(1):100–107
2. Park JH, Lee KH, Lee JW, Lim SB (2015) Design of voice annotation system to enhance the quality of life for visually-impaired readers. *Asian Int J Life Sc, ASIA LIFE SCI Suppl* 11:109–121
3. Seo SH, Lee JW, Lim SB (2012) A reading technique of math expression in e-book for reading-disabled people. *J HCI Soc Korea* 7(2):57–64
4. W3C, Mathematical Markup Language (MathML) Version 3.0 2nd Edition, 2014
5. W3C, XSLT Specification, Available: <https://www.w3.org/TR/xslt> Accessed 30 March 2018
6. DAISY Consortium (2005) Specifications for the Digital Talking Book, ANSI/NISO Z39.86–2005
7. Neil Soiffer (2015) Browser-independent accessible math, in *Proceeding of W4A 2015*, Article No. 28
8. gh-MathSpeak. MathSpeak Core Specification Grammar rules, Available: <http://www.gh-mathspeak.com/examples/grammar-rules/> Accessed 15 March 2018
9. MathGenie, MathGenie, Available: <http://www.mathgenie.com/> Accessed 15 March 2018
10. Wiris, MathType, Available: <http://www.wiris.com/mathtype> Accessed 25 May 2020
11. Davide C, Peter K, Volke S (2016) New accessibility features. *J Technol Persons Disabil* 4:167–175
12. Dumkasem K, Srisingchai P, Rattanatamrong P (2019) EyeMath: Increasing Accessibility of Mathematics to Visually Impaired Readers, In: *Proceeding of 2019 23rd International Computer Science and Engineering Conference (ICSEC)*, pp. 197–202
13. Kacorri H, Riga P, Kouroupetroglou G (2014) EAR-Math: Evaluation of Audio Rendered Mathematics, In: *Proceeding of UAHCI 2014: Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge*, pp. 111–120
14. Fitzpatrick D (1999) *Towards Accessible Technical Documents: Production of Speech and Braille Output from Formatted Documents*. Doctoral dissertation, Dublin City University
15. Yamaguchi K, Suzuki M (2010) On necessity of a new method to read out math contents properly in DAISY, In *Proceeding of International Conference on Computers for Handicapped Persons 2010(ICCHP)*, 415–422

16. Wongkia W, Naruedomkul K, Cercone N (2009) Better Access to Math for Visually Impaired, In Proceeding of 2009 IEEE Toronto International Conference Science and Technology for Humanity , 43–48
17. Salamonczyk A, Brzostek-Pawlowska J (2015) Translation of MathML formulas to Polish text, example applications in teaching the blind in Proceeding of IEEE 2nd International Conference on Cybernetics (CYBCONF), 240–244
18. W3C, SSML Specification, Available: <http://www.w3.org/TR/speech-synthesis/> (accessed March, 15, 2018)
19. Tullis T, Alvert B (2009) Measuring user experience. Elsevier, Amsterdam

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.