독서 장애인 용 전자책 내 수식 독음 기법

A Reading Technique of math expression in e-Book for Reading-disabled People

서승희, SeungHee Seo*, 이종우, Jongwoo Lee**, 임순범, Soon-Bum Lim***

요약 최근 독서 장애인을 위한 국제 디지털 음성 도서 표준인 DAISY를 통해 장애인의 고등 교육을 위한 연구가 활성화되고 있으나 그림이나 수식과 같은 특수요소는 음성도서에서 독음해주기 힘들다. 이를 위해, 본 연구에서는 중학교 및 고등학교 1학년 교과과정에 포함된 수식을 토대로 Contents MathML에 대한 한글 독음 규칙을 정의하였다. 또한, 정의한 한글 독음 규칙에 따라 Contents MathML로 표현된 수식을 독음해주기 위한 프로그램을 XSLT로 구현하였다. 본 연구에서 구현한 프로그램을 검증하기 위해 사용성을 검토하였다.

Abstract Digital talking books have been developed to enhance reading experiences for reading-disabled people. In the existing digital talking books, however, it is difficult to read special contents such as math expressions, images, etc. In this paper, we define reading rules for Contents MathML. Based on this rules, we implemented a program for reading the mathematical contents of Contents MathML, using XSLT. We conducted a confirm test to prove the accuracy of the program, and an usability test to measure understanding of reading text of math expressions.

***핵심어:** Reading disables, MathML, Math to speech*

본 논문은 숙명여자대학교의 2012학년도 교내 연구비 지원에 의해 수행되었음.
*주저자 : 숙명여자대학교 멀티미디어과학과 석사과정
**공동저자 : 숙명여자대학교 멀티미디어과학과 교수
***교신저자 : 숙명여자대학교 멀티미디어과학과 교수; e-mail: sblim@sookmyung.ac.kr
■ 접수일 : 2012년 9월 24일 / 심사일 : 2012년 10월 15일 / 게재확정일 : 2012년 10월 23일

1. 서론

독서 장애인의 독서 환경 개선을 위해 다양한 디지털 음성 도서 기술이 개발되고 있다. 디지털 음성도서란 시각 장애인, 난독증을 포함한 독서 장애인을 위해 도서 녹음 파일이나 컴퓨터 합성음으로 책을 읽어주는 도서를 말한다. 이러한 디지털 음성도서 기술 중 DAISY가 국제 디지털 음성도서의 표준으로 널리 사용되고 있다. 이러한 디지털 음성도서의 개발과 함께 독서 장애인의 교육 환경 개선을 위한 다양한 전자 교과서 연구가 활성화되고 있다. DAISY를 통한 텍스트 읽어주기 기능은 표준화되어 많은 독서장애인, 저시력자가 사용하고 있지만 도서에 포함되는 그림이나 수식 등의 특수 콘텐츠의 독음에 대한 연구가 제대로 이루어지지 않고 있다. 특히 DAISY 내의 수식 콘텐츠의 독음에 대한 국내외 연구가 매우 미흡한 상태이다.

DAISY 내의 수식 콘텐츠는 웹에서도 널리 사용되고 있는 MathML로 표현된다. 본 연구에서는 MathML 콘텐츠의 분석과 중학교 교과 과정에 포함된 수식을 대상으로 MathML로 표현된 수식 콘텐츠의 독음을 위해 수식 독음 규칙을 정리하고, 정리한 독음 규칙을 적용한 SLT 프로그램을 구현하여 독음 텍스트를 추출하는 내용을 주로 다루고 있다[1].

본 연구에서는 음성 전자책에서 MathML로 표현된 수식 독음 기법을 개발하였다. 이 중 의미 표현이 가능한 Contents MathML에 대한 독음 규칙을 정의하고, 플러그인 형태가 아니라 XSLT[2]를 사용하여 확정성을 높이고 처리 속도가 빨라지도록 하였다. 수식의 범위는 중학교 교과과정부터 고등학교 1학년에 해당하는 공통 수학 과정까지 등장하는 수식을 대상으로 독음 기법을 연구하였다.

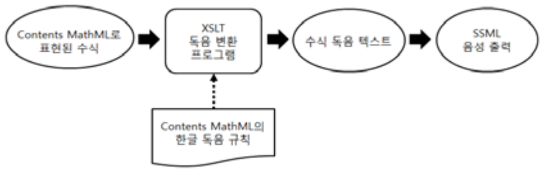

그림 1. 수식 콘텐츠의 독음 프로그램 시스템

2. 관련 연구

디지털 음성 도서 표준인 DAISY 내에서 수식 콘텐츠는 MathML로 표현되고 있다. 연구의 이해를 돕기 위해 MathML에 대해 알아본다. 그리고 국외에서 이루어지고 있는 수식 독음 서비스인 MathPlayer와 국내 본 연구진에서 선행한 일반 문서 편집기 내의 수식 편집기에서 사용되는 수식 표현법을 읽어주기 위한 연구인 MathExpression Reader에 대해 알아본다.

2.1 MathML

MathML은 XML의 응용 중 하나로 수학 수식을 표현하기 위한 마크업 언어이며 수학 수식을 표현하는 웹 표준으로 자리잡고 있다. MathML은 단순한 수식의 표기 뿐만 아니라 수식의 의미도 표현이 가능하여 MathML로 표현된 수식 콘텐츠를 기계가 이해할 수 있으며 타 응용 프로그램 간의 의미 교환이 가능하다. 이러한 MathML은 Presentation MathML과 Contents MathML로 나뉘어진다[3].

2.1.1 Presentation MathML

Presentation MathML은 수학 수식을 웹 브라우저 상의 표현을 중점적인 목표로 만들어진 마크업 언어이다. Presentation MathML은 레이아웃에 따라 요소가 정의된다. 현재 약 30개의 요소와 50개의 속성 값이 정의되어 있다[4].

2.1.2 Contents MathML

Contents MathML은 수식의 레이아웃 보다 수식의 의미에 집중하여 명확한 인코딩이 가능하게 하는 마크업이다. 따라서 Contents MathML로 표현된 수식은 Presentation MathML로 표현된 수식보다 모호성이 적다고 할 수 있다. 숫자나 변수를 표현할 때는 Presentation MathML과 비슷한 방식으로 ci나 cn 이라는 요소를 사용하지만, 연산자를 표현하는 노드는 수학적 의미를 담은 특별한 노드로 구분된다[5].

2.2 MathPlayer

해외의 수식 독음 서비스로 DesignScience의 MathPlayer가 있고 이에 적용된 MathSpeak의 문법 규칙이 있다. MathPlayer는 MathML을 웹 브라우저에서 볼 수 있게 렌더링하고, 이 수식을 읽어주는 서비스를 제공하고 있다. MathPlayer는 플러그인의 설치 후 Internet Explorer 브라우저에서 사용 가능하다[6]. 이에 적용된 문법 규칙은 2004년에 시작된 gh-MathSpeak사의 MathSpeak 프로젝트를 통해 고안된 독음 스타일인 MathSpeak이다. MathSpeak의 문법 규칙은 충분한 독음(verbose), 중간 단계(brief), 간략 단계(super-brief)의 세 가지 단계의 독음 스타일을 제공하여 청자의 상황이나 요구조건에 따라 독음 단계를 선택하여 수식을 들을 수 있도록 한다[7].

2.3 MathExpression Reader

국내에서는 일반 문서 편집기 내의 수식 편집기에서 사용되는 수식 표현법을 독음텍스트로 추출하는 연구를 본 연구진에서 선행하였다. 이 연구는 중학교 교과과정에 포함되는 수식을 대상으로 하였다. 이 연구에서는 국내의 환경에 맞게 수식 독음 규칙을 정리하고, 정리된 독음 규칙에 따라 수식 편집기의 식에서 독음 텍스트를 추출하는 프로그램을 구현하였다[8].

3. 수식 콘텐츠 독음 변환 기법

3.1 Contents MathML의 노드 트리 분석

Contents MathML은 의미를 표현하여 전달하기 위한 MathML로써 기계가독 및 계산을 위한 수식 표현 방식이다. 그림 2와 같이 하나의 수식은 〈apply〉라는 노드로 묶이게 되고 〈apply〉의 첫 번째 자식 노드는 연산자 노드, 그다음에 오는 노드는 피연산자 노드이다.

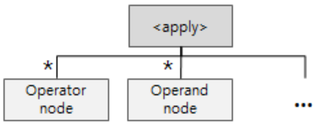

그림 2. Contents MathML의 노드 트리 구조

피연산자 노드가 1개일때는 대부분 우피연산자이며, 피연산자 노드가 2개일 때는 좌피연산자, 우피연산자 순으로 노드가 오게 되며, 피연산자 노드의 수가 3개 이상일 때는 하나의 연산자 노드가 여러 피연산자가 반복되어 나오는 경우이다. 예를 들면, a+b+c+d의 경우 +라는 연산자로 묶이는 경우이므로 연산자 노드는 〈plus〉가 되며 피연산자 노드의 수는 4개가 된다.

```
...
〈math〉
〈apply〉
  〈plus/〉
  〈ci〉 a 〈/ci〉 〈ci〉 b 〈/ci〉 〈ci〉 c 〈/ci〉 〈ci〉 d 〈/ci〉
〈/apply〉
...
```

그림 3. (a+b+c+d)를 표현한 Contents MathML

3.2 Contents MathML 수식 독음 규칙

Contents MathML의 노드 구성을 수식의 특징에 따라 크게 메타 엘리먼트, 연산자 엘리먼트, 피연산자 엘리먼트로 분류하였고, 이에 따른 독음 규칙을 정의하였다.

3.2.1 메타 엘리먼트

메타 엘리먼트는 MathML의 루트 엘리먼트에 해당하는 〈math〉와 수식의 부가정보를 표현하는 Semantic Mapping 엘리먼트로 이루어진다.

Contents MathML은 〈math〉로 시작되는 식 트리로 수식을 표현한다. 〈math〉는 루트 엘리먼트에 해당하므로 의미를 가지지 않아 따로 한글 독음으로 바꿔주지 않는다.

Semantic Mapping 엘리먼트는 수식의 부가정보를 표현하는 노드이다. Semantic Mapping 엘리먼트에 해당하는 노드는 〈annotation〉, 〈semantics〉, 〈annotation-xml〉이 있으며, 연산자 노드와 피연산자 노드의 분석을 통해 수식 독음이 가능하므로 세 노드에 대해서는 한글 독음으로 바꾸지 않는다.

3.2.2 피 연산자 엘리먼트

수식의 피연산자를 표현하는 엘리먼트로, 〈apply〉의 나머지 자식 요소에서 첫 번째 자식 요소인 연산자에 대해 피연산자 엘리먼트가 쓰인다. 토큰 요소가 단말 노드로 오거나 〈apply〉 요소가 형성한 식 트리가 중첩되어 올 수 있다.

토큰 요소는 수(Numbers), 변수(Content Identifiers), 상수 및 심볼 요소(Constant and Symbol elements)로 구성된다. 첫째, 수를 나타내는 〈cn〉은 노드의 텍스트 값을 그대로 출력한다. 다만, 〈cn〉의 텍스트 값이 다음의 심볼 값 중 하나일 때, 다음의 독음 규칙을 적용한다.

표 1. 〈cn〉 독음 시 심볼에 대한 독음 규칙

노드	독음
value(symbol)	독음
π	파이
ⅇ (or ⅇ)	익스포넨셜 e
ⅈ (or ⅈ)	허수 i
γ	감마
∞ (or &infty;)	무한대

둘째, 변수를 나타내는 〈ci〉는 한글 독음으로 바꿔주지 않고 그 값을 그대로 출력한다. 본 노드에서 type이라는 속성을 사용했을 시, 속성에 따라 변수의 정보를 얻을 수 있으므로 type 속성의 값에 따라 다음의 단어를 추가하여 독음한다.

표 2. 〈ci〉의 type 속성 값에 따른 독음 규칙

type 속성 값	독음	type 속성 값	독음
integer	정수	constant	상수
rational	유리수	function	함수
real	실수	vector	벡터
complex	복소수	set	집합

피연산자의 비단말 노드에 해당하는 〈apply〉는 중첩하여 사용할 수 있다. 다음 장에 언급한 경우를 제외한 나머지는 중첩된 〈apply〉에 '괄호 열고'와 '괄호 닫고'를 수식의 전후에 매핑하여 연산자의 범위를 명확하게 한다.

표 3. 중첩된 〈apply〉의 독음 규칙

〈apply〉 중첩 표현	독음
〈apply〉 *operator argument(s)* 〈/apply〉	괄호 열고 *operator argument(s)* 괄호 닫고

3.3.3 연산자 엘리먼트

수식의 연산자를 표현하는 엘리먼트로, 〈apply〉의 첫 번째 자식 요소에 연산자 엘리먼트가 등장한다. 연산자 엘리먼트 중 중등 3년 교과 과정과 고교 공통 수학에 등장하는 수식 연산자를 대상으로 독음 규칙을 정의하였다. 연산자는 크게 기본요소, 산술 논리 대수, 관계, 집합 이론, 기본 고전 함수로 분류하였다. 그에 따른 독음 규칙은 다음과 같다.

다음 표 4에 나오는 표현 중 a, b는 단말노드를, [a], [b]는 비 단말노드에 해당한다. 단말노드는 노드를 그대로 독음하며, 비단말 노드의 경우에는 앞서 정의한 기본 독음 규칙인 '괄호 닫고', '괄호 열고'를 추가하여 독음한다.

4. 구현 및 구현 결과

앞서 정리한 독음 규칙을 토대로 Contents MathML을 위한 독음 프로그램을 XSLT로 구현하였다. Contents MathML로 표현된 수식을 XSLT가 읽어들여 각 연산자에 맞는 독음 텍스트 변환 템플릿을 호출하여 독음텍스트로 변환하게 한다. Contents MathML로 표현된 수식이 독음 텍스트로 변환을 모두 마치게 되면, 추출된 독음 텍스트에 음성 스타일 적용 템플릿을 적용하여 SSML 파일을 출력하게 된다. 이를 통해 Contents MathML 콘텐츠에 대해 음성지원을 할 수 있게 한다.

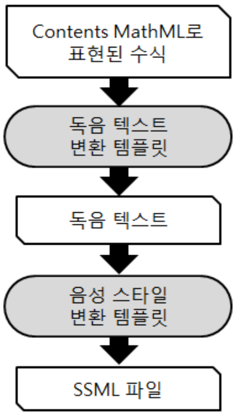

그림 4. 독음 프로그램 모듈

표 4. 연산자에 대한 독음 변환 규칙

	Contents MathML 표현방식	수학기호	한글 독음
기본 요소 Basic Element	〈fn〉 f 〈/fn〉[x]	f(x)	함수 f 괄호열고 [x] 괄호닫고
	〈inverse/〉	A^{-1}	역
	〈compose/〉[g] [f]*	$f \circ g$	[g] [합성 f]*
산술 논리 대수 Arithmetic, Algebra and Logic	〈divide/〉 a b	$\frac{a}{b}$	분수시작 b분의 a 분수 끝
	〈divide/〉 [a] [b]		분수시작 분모 [b] 분모 끝 분의 분자 [a] 분자 끝
	〈max/〉 [a] [b] [n]*	max {a, b, n*}	[a]와/과 [b](와/과 [n])*의 최대값
	〈min/〉 [a] [b] [n]*	min {a, b, n*}	[a]와/과 [b](와/과 [n])*의 최소값
	〈minus/〉 [a]	−a	마이너스 [a]
	〈minus/〉 [a] [b]	a−b	[a] 마이너스 [b]
	〈plus/〉 [a] [b] [n]*	a+b(+n)*	[a] 플러스 [b] (플러스 [n])*
	〈power/〉 a b	a^b	a의 b승
	〈power/〉 [a] [b]		[a]의 지수 [b]지수 끝 승
	〈times/〉 [a] [b] [n]*	ab(n)*	[a] 곱하기 [b] (곱하기 [n])*
	〈root/〉 [b]	$\sqrt{b}$	루트 시작 [b] 루트 끝
	〈root/〉 〈degree〉 [a] 〈degree〉 [b]	$\sqrt[a]{b}$	루트 시작 [b] 의 [a] 제곱근 루트 끝
	〈gcd/〉 [a] [b] [n]*	gcd (a, b, n*)	[a]와/과 [b] (와/과 [n])*의 최대공약수
	〈abs/〉 [a]	|a|	절대값 시작 [a] 절대값 끝
	〈lcm/〉 [a] [b]	lcm (a, b, n*)	[a]와/과 [b] (와/과 [n])*의 최소공배수
	〈factorial/〉 [A]	A!	[A] 팩토리얼
	〈implies/〉[P] [Q]	$p \Rightarrow q$	[P] 이면 [Q]이다

	Contents MathML 표현방식	수학기호	한글 독음
관계 Relations	〈eq/〉 [a] [b] [n]*	a=b(=n)*	[a]은/는 [b](이고 [n])* 이다
	〈neq/〉 [a] [b]	a!=b	[a]은/는 [b] 아니다
	〈gt/〉 [a] [b] [n]*	a〉b(〉n)*	[a]은/는 [b]보다 (크고 [n−1]은/는 [n]보다)* 크다.
	〈lt/〉 [a] [b] [n]*	a〈b(〈n)*	[a]은/는 [b]보다 작고 [n−1]은/는 [n]보다)* 작다.
	〈geq/〉 [a] [b] [n]*	a≥b≥(n)*	[a]은/는 [b]보다 (크거나 같고[n−1]은/는 [n]]보다)* 크거나 같다
	〈leq/〉 [a] [b] [n]*	a≤b≤(n)*	[a]은/는 [b]보다 (작거나 같고 [n−1]은/는 [n]보다)* 작거나 같다
집합 이론 Theroy of Sets	〈set〉 [a] [b] [n]*	set{a, b, n*}	집합 [a] [b] ([n])*
	〈union/〉 [A] [B] [n]*	A∪B(∪n)*	[A] 합집합 [B] (합집합 [n])*
	〈intersect/〉 [A] [B] [n]*	A∩B(∩n)*	{A} 교집합 [B] (교집합 [n])*
	〈in/〉 [a] [A]	a ∈ A	[a] 은/는 [A]의 원소이다
	〈notin/〉 [a] [A]	a ∉ A	[a] 은/는 [A]의 원소가 아니다
	〈prsubset/〉 [A] [B]	A ⊂ B	[A]은/는 [B]의 부분 집합이다
	〈notprsubset/〉 [A] [B]	A ⊄ B	[A]은/는 [B]의 부분집합이 아니다
기본 고전 함수 Elementary Classical Functions	〈sin/〉 [x]	$\sin x$	사인 [x]
	〈cos/〉 [x]	$\cos x$	코사인 [x]
	〈tan/〉 [x]	$\tan x$	탄젠트 [x]
	〈log/〉〈logbase〉 A〈/logbase〉 B	$\log_a b$	로그 시작 A의 B 로그 끝
	〈log/〉〈logbase〉 [A] 〈/logbase〉 [B]		로그 밑 시작 [A] 밑 끝의 [B] 로그 끝

*단말노드: 〈cn〉이나 〈ci〉노드/ 비단말 노드: 〈apply〉나 〈reln〉
[a], [A]: 비단말노드 a, A: 단말노드 A : 집합 a : 원소

Contents MathML로 표현된 수식은 연산자 노드가 〈apply〉 노드의 첫 번째 자식으로 등장하고 피연산자 노드가 연산자 노드의 형제노드로 등장한다. 이러한 노드 트리의 순서를 독음 규칙에 맞춘 순서로 매핑해 줄 필요가 있다. 노드 트리 정규화 모듈에서는 앞서 정의한 독음 규칙을 적용하여 수식에 따라 독음 규칙에 맞는 템플릿을 매핑한다.

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

그림 5. 〈apply〉 노드에의 템플릿 적용

Contents MathML에서의 연산자 노드는 두 가지 특징으로 나눌 수 있는데 형제 노드의 유무, 자식 노드의 유무에 따라 나눌 수 있다. 우선적으로 자식 노드의 유무를 판단하여 자식 노드의 독음 템플릿을 적용한 후 형제 노드를 확인한다. 형제노드를 처리할 때는 형제 노드의 수에 따라 독음 템플릿을 판단한다. 형제노드가 없는 단일 노드일 경우에는 피연산자 노드 중 단말 노드이므로 단말 노드에 해당하는 템플릿을 매핑한다. 형제노드가 하나일 경우에는 좌피연산자만 가지는 노드이므로 연산자 노드를 먼저 매핑한 후 피연산자 노드의 템플릿을 매핑한다. 형제노드가 2개일 경우에는 연산자의 좌/우 피연산자가 존재하는 경우로, 국내의 독음 규칙에 따라 연산 순서를 정한 후 노드에 맞는 템플릿을 호출한다. 형제 노드가 3개 이상일 경우에는 하나의 연산자가 반복되어 나오는 수식의 경우이다. 사칙 연산과 같은 연산자가 반복될 경우, 예를 들어 a * b * c * d라면, a 곱하기 b 곱하기 c 곱하기 d 와 같이 연산자만 반복하여 템플릿을 호출하면 된다. 하지만 부등호의 경우, 예를 들어 a〉b〉c라면 연산자 템플릿만 호출한다면 'a는 b보다 크고 c보다 크다'로 독음 되어 의미가 모호해진다. 이를 위해 피연산자의 반복 호출을 통해 의미의 모호성을 제거한다.

독음 텍스트 추출 모듈에서는 각 노드에 따른 독음 텍스트를 추출하고 이를 합성하여 전체 수식의 독음 텍스트를 추출하는 모듈이다. 이 때, 피연산자 노드의 마지막 자의 종성의 유무에 따라 조사를 판단하여 독음 텍스트를 추출한다. 종성이 있을 경우에는 은/는 중 '은', 이/가 중 '이', 을/를 중 '을'을 선택하고 종성이 없을 경우에는 나머지를 택한다. 또한 피연산자의 복잡도에 따라 그룹핑 독음을 추가하는데, 연산자의 종류에 따라 추가되는 그룹핑 독음을 달리한다.

```
〈xsl:template match="divide"〉
  〈xsl:apply-templates select="following-sibling::*[2]"〉
    〈xsl:if test=".[name()='ci']"〉
      〈xsl:apply-templates select="."/〉
```

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

그림 6. 연산자 노드에의 템플릿 적용 예시

추출된 독음 텍스트를 읽어주기 위해서 독음 텍스트로 SSML 콘텐츠를 생성한다[10].

이를 통해 구현한 결과는 다음 그림 7과 같다.

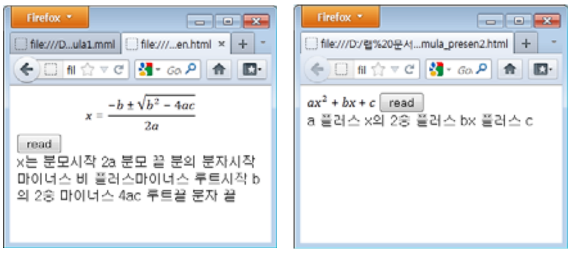

그림 7. 실행 프로그램 화면

5. 사용성 평가

5.1 적합성 검사

중학교 수학 교과과정 및 고등학교 교과 과정 중 공통 수학에 해당하는 범위의 수식 예제들을 본 논문에서 구현한 독음 프로그램으로 모두 변환해 보았다. 그 결과 모든 수식 및 예제들을 앞서 보여준 예제의 실행 사례와 동일하게 한글 수식 독음으로 변환됨을 확인할 수 있었다. 생성된 수식은 다음 표 5와 같다.

표 5. 각 영역별 실행 사례

범위	대표 수식 예제	수식 예제의 한글 독음
기본 요소	$f(x)=3x+4$	함수 f 괄호열고 x 괄호닫고 는 3 x 플러스 4
	$f \circ g$	f 합성 g
	…	…
산술 논리 대수	$\frac{a+b}{c+3}=20$	분모시작 c 플러스 3 분모 끝 분의 분자 시작 a 플러스 b 분자 끝 은 20
	$\sqrt{a+b+2}$	루트 시작 a 플러스 b 플러스 2 루트 끝
	…	…
관계	$3a < b < \frac{c}{2}$	3a는 b보다 작고 b는 2 분의 c 보다 작다
	…	…
집합 이론	$A \cup B = B \cup A$	A 합집합 B 는 B 합집합 A
	…	…
기본 고전 함수	$b = c\sin B$	b 는 c 사인 B
	…	…

5.2 독음의 이해도 검사

제안된 독음 프로그램에서 수식 독음의 이해도를 측정하기 위해 사용성 평가를 실시하였다. 사용자 5명을 대상으로 3일 간격으로 독음의 속도를 달리하여 수식 받아쓰기를 실시하였다. 난이도에 따라 4문제 씩 출제하였고, 받아 쓴 수식의 완성도에 따라 점수를 매겼다.

표 6. 실험 방법 및 측정 방법

실험 방법	1) 난이도 상, 중, 하에 따라 각 5개 씩 총 15개의 수식을 듣고 받아쓰게 한다. 2) 일반적으로 책을 읽는 속도보다 30% 빠른 속도로 수식을 읽어주어 받아쓰게 한다. 3) 1차 테스트 3일 뒤에는 일반적으로 책을 읽는 속도로 수식을 읽어주어 받아쓰게 한다. 4) 2차 테스트 3일 뒤에는 일반적으로 책을 읽는 속도보다 30% 정도 느린 속도로 수식을 읽어주어 받아쓰게 한다. * 모든 실험은 1회만 들려준다.
측정 방법	1) 받아 쓴 수식과 원래 수식을 비교하여 맞춘 개수를 센다. 2) 난이도에 따라 점수의 평균을 매긴다.

비슷한 내용의 수식으로 같은 집단에 독음의 속도를 다르게 하여 들려주었다. 독음의 속도는 전자책 뷰어에서의 독음 속도인 평소 책을 읽는 속도보다 30% 정도 빠른 속도, 평소 책을 읽는 속도, 그리고 평소 책을 읽는 속도보다 30% 느린 속도로 세 가지의 속도로 수식을 독음하였다. 들려 준 독음을 토대로 받아쓰게 하여 원래의 수식에의 정확도를 측정하여 이해도를 판단하였다. 측정 결과를 토대로 각 문항의 난이도를 상, 중, 하로 나누어 문항 별 점수의 평균을 도출하였다. 실험 결과는 다음 표 7과 같다.

표 7. 수식 받아쓰기 실험 결과

조건 1	평소 책을 읽는 속도 보다 30% 빠르게 독음 했을 때 (전자책 뷰어에서의 독음 속도)					
	사용자1	사용자2	사용자3	사용자4	사용자5	평균
하	4	4	4	4	4	4
중	1	0	0	1	0	0.4
상	1	0	0	0	0	0.2

조건 2	평소 책을 읽는 속도로 독음 했을 때 (보통)					
하	4	4	4	4	4	4
중	1	0	1	2	1	1
상	1	1	1	0	1	0.8
조건 3	평소 책을 읽는 속도보다 30% 느리게 독음 했을 때 (느리게)					
하	4	4	4	4	4	4
중	2	1	2	3	2	2
상	1	1	1	2	2	1.4

위의 결과를 보았을 때, 수식의 복잡도가 클 경우, 즉 독음의 길이가 길 경우에는 알아듣기 힘들며 맞추는 수가 하락하는 것을 볼 수 있다. 또한 독음의 속도를 빠르게 할수록 이해도가 낮아지는 것을 맞춘 수의 평균이 낮아지는 것을 통해 알 수 있다.

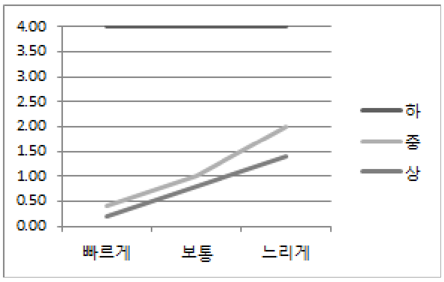

그림 8. 수식 난이도와 독음 속도에 따른 평균 그래프

각 난이도와 독음 속도에 따른 평균을 그림 8과 같은 그래프로 나타내 보았다. 난이도 '하' 수준의 수식은 독음의 속도가 빠르더라도 모두 이해할 수 있는 것으로 나타났다. 하지만 난이도 '중' 이상이 되었을 때, 독음 속도가 빨라질수록 수식의 이해도가 현저히 하락함을 보였다. 이를 통해, 아주 간단한 수식은 빠른 속도로 독음하여도 수식의 이해에 문제가 없으나, 수식의 난이도가 높아질수록 수식 독음을 느리게 해야 함을 알 수 있다. 또한 테스트 진행과 함께 공통적으로 피 실험자들이 제안한 의견을 통해 끊어 읽기 등의 수식 독음 방식에 대한 연구가 추후 필요한 것으로 보인다.

5. 결론 및 제언

본 연구는 독서 장애인의 수식 콘텐츠에 대한 접근성을 높이기 위한 연구로서 장애인용 전자책 내에서 Contents MathML로 표현되는 수식 콘텐츠의 독음 프로그램을 구현하였다. 중학교 교과 과정부터 고등학교 1학년에 해당하는 공통 수학까지 포함되는 수식의 독음규칙 정리와 프로그램 구현과 적합성 검사를 통해 본 연구에서 구현한 프로그램이 수식을 제대로 변환함을 확인하였다.

사용성 평가 결과, 독음의 속도가 수학 수식의 이해도에 영향을 주는 것을 확인하였다. 이에 따라 독음 속도나 끊어 읽기 등의 독음 방식에 대한 추가 연구를 수행할 예정이며 향후 Presentation MathML의 독음 텍스트 추출 프로그램을 구현할 예정이다.

참고문헌

[1] DAISY Consortium, Specifications for the Digital Talking Book http://www.daisy.org/z3986/2005/Z3986-2005.html?q=z3986/2005/z3986-2005.html

[2] W3C, XSLT Specification, http://www.w3.org/TR/xslt

[3] W3C, MathML Specification, http://www.w3.org/TR/MathML2

[4] W3C, Presentation MarkupSpecification, http://www.w3.org/TR/MathML2/chapter3.html

[5] W3C, Content MarkupSpecification, http://www.w3.org/TR/MathML2/chapter4.html

[6] Design Science, MathPlayer Test Suite http://www.dessci.com/en/reference/MathMLTestSuite/index.html

[7] gh-MathSpeak, MathSpeak Core Specification Grammar rules http://www.gh-mathspeak.com/examples/grammar-rules/

[8] 이재화, 이종우, 임순범. 국내 음성 도서 서비스를 위한 수식의 음성변환 기법에 대한 연구. 한국정보과학회 2011 한국 컴퓨터 학술 발표논문집 제38권 제1호(B). pp. 226-227. 2011.

[9] W3C, Speech Synthesis Markup Language. http://www.w3.org/TR/speech-synthesis/

[10] 이경희, 이종우, 임순범. 디지털 말하기 책을 위한 음성 주석 달기 시스템. 정보과학회논문지. 컴퓨팅의 실제 및 레터 제18권 제4호. pp. 276-281. 2012.

서 승 희

2007년 3월~2011년 2월 숙명여자대학교 멀티미디어과학과 졸업(공학사). 2011년 3월~현재 숙명여자대학교 대학원 멀티미디어과학과 석사과정. 관심분야는 전자출판, User interface임

이 종 우

1986년 3월~1990년 8월 서울대학교 컴퓨터공학과 졸업(공학사). 1990년 3월~1992년 8월 서울대학교 대학원 컴퓨터공학과 석사 졸업(공학석사). 1992년 3월~1996년 8월 서울대학교 대학원 박사 졸업 (공학박사).

1992년 3월~1998년 6월 (주)현대전자산업 정보시스템 사업부 과장. 1998년 7월~1999년 7월 (주) 현대정보기술 기술본부 책임연구원. 1999년 9월~2002년 8월 한림대학교 정보통신공학부 조교수. 2002년 9월~2003년 8월 광운대학교 컴퓨터공학부 조교수. 2003년 9월~2004년 7월 (주)아이닉스 소프트 개발이사. 2004년 9월~현재 숙명여자대학교 정보과학부 멀티미디어과학과 부교수.

관심분야는 Multimedia Storage System, Multimedia Embedded System, Multimedia Operating System임.

임 순 범

1977년 3월~1982년 2월 서울대학교 계산통계학과 졸업. 1982년 3월~1983년 8월 한국과학기술원 전산학과 석사 졸업. 1983년 9월~1992년 2월 한국과학기술원 전산학과 박사 졸업.

1989년 2월~1992년 2월 (주) 휴먼컴퓨터 창업(연구소장). 1992년 3월~1997년 2월 (주) 삼보컴퓨터 부장(프린터개발부). 1997년 3월~2001 8월 건국대학교 컴퓨터과학과 교수. 2001년 9월~현재 숙명여자대학교 정보과학부 멀티미디어과학과 교수.

관심분야는 컴퓨터그래픽스, 웹/모바일 멀티미디어 응용, 디지털 방송, 전자출판, User Interface임.