

## 비점역자를 위한 ‘음악점자 변환기’ 설계 및 구현 Design and Implementation of Music-To-Braille Translator

남윤건\*, 민흥기

Y. K. Nam, H. K. Min

### 요 약

음악 점자는 문자 점자에 비해 보다 체계적이고 다양한 기호들로 구성되어 있다. 그러므로 음악 점역은 점자 기호의 숙지와 함께 악보에 대한 해박한 지식이 필요하다. 현재 문자 점역은 컴퓨터 프로그램이 개발되어 있어서 점역사의 도움 없이 점역을 할 수 있다. 하지만 음악 점역은 아직까지 음악 전문 점역사의 도움 없이 컴퓨터 프로그램으로 자동 점역하는 것이 어렵다. 음악 점역사가 악보를 보고 직접 점자로 작성하는 실정이다. 본 논문은 점자를 잘 이해하지 못하는 사람도 음악점역을 할 수 있는 ‘음악점자 변환기’를 설계 및 구현 하였다. 가사처리를 위해 문자 점역을 포함하고 있으며, 악보 구현은 음표, 쉼표, 길표, 조표, 박자표, 타이, 슬러, 도돌이표를 개발하여 실제 악보의 성공적 변환을 확인하였다.

### ABSTRACT

Music braille is made up of more systematic and various symbols compared with text braille. Therefore, the translation of braille music requires extensive knowledge of the music and the braille symbols. Because currently the computer program for the text braille has already developed, we don't need any help from the braille translator. However, the translation of music braille is hard without the help of a professional music braille translator, because the translation computer program is not perfect. The current situation is that the music braille translator reads the music score and translates it into the braille himself. In this paper, we designed and implemented the "Braille Music Converter", you can implement a person does not understand the braille translation into music braille well. It includes translation into text braille for the lyrics processing and rest, octave, key signature, time signature, tie, slur, repeat mark was confirmed that the successful conversion to the actual music score.

Keyword : Braille Translator, Accessibility, Music Braille, Printed Letter

### 1. 서론

시각장애인이 손으로 읽는 글을 ‘점자’라고 하고 정상인이 눈으로 읽는 글을 ‘묵자(printed letter)’라고 하며, ‘묵자’를 ‘점자’로 변환하는 것을 ‘점역’이라고 한다. 점자는 수백 년 전에 개발되어 발전되어 왔으며, 오늘날 6점형 점자체계가 널리 사용되고 있다.[1]

한글의 경우는 ‘점사랑’이라는 점역 프로그램이 개발되어 있고, IT 발달로 많은 책들이 전자적으로 출판되어 있어서, 프로그램을 이용하면 점역에 큰 어려움이 없다. 하지만, 음악이나 수학, 과학처럼 다양한 기호로 이루어진 문서의 경우는 자동으로 점역이 이루어지기 까다롭다. 그러므로 실제 점역사 자격증을 보유한 분들의 수요는 대부분 음악, 수학, 과학과 같은 전문분야이다. 특히 음악의 경우는 다양한 편곡이 계속되고 있고, 점역이 까다로워 수요에 공급이 턱없이 부족하다. 음악 점자는 다른 어떤 분야의 점자들보다도 체계적이고, 다양한 기호들로 구성되어 있으며, 음악 점역은 점자 기호의 숙지와 더불어 묵자 악보에 대한 충분한 이해와 지식이 요구되기 때문에 음악 점역을 능숙하게 구현하기 위해서는

접 수 일 : 2016.07.01

심사완료일 : 2016.07.21

게재확정일 : 2016.07.25

\* 남윤건 : 서울 보인고등학교 학생  
imygnam@gmail.com (주저자)

민흥기 : 인천대학교 정보통신공학과 교수  
hkmin@inu.ac.kr (교신저자)

수많은 시간의 연습과 노력이 필요하다.

현재 음악점자 변환 프로그램으로는 덕스베리(Duxbury), 굿필(GOODFEEL)이 개발되어 있으나 모두 자동변환기능이 약하여 전문 음악점역사의 도움 없이는 점역이 어렵다.[2, 3, 4]

본 논문은 점자를 잘 이해하지 못하는 사람도 음악점역을 구현할 수 있는 ‘음악점자 변환기’를 설계 및 구현하였다. 구현된 프로그램은 입력부분은 음악 기호를 선택하여 처리하였으며, 이를 점자로 번역하고 목자악보로 표현하는 기능으로 구현하였다. 개발된 프로그램을 통해 실제 악보를 입력하여 음악 점역사의 검증을 통해 성공적으로 악보의 점역이 이루어졌음을 확인하였다.

## 2. 관련연구

현재 가장 많이 사용하는 점역프로그램으로는 ‘점사랑’이라는 소프트웨어가 있다. 점사랑은 문자를 점자로 변환하는 기능과 함께 바로 점자를 입력하는 기능으로 구성되어 있다. 악보를 점역하기 위해서는 음악점역의 이해를 바탕으로 점역사가 바로 점자로 입력하는 기능을 이용하여 처리하고 있다. 이 때문에 음악 점역사는 악보를 점자로 표현하기 위한 다양한 기술을 보유하고 있어야 한다.

### 2.1 가사 처리를 위한 한글점역

악보에는 제목과 빠르기, 저자, 가사 등 기본적으로 문자가 포함되어 있다. 문자를 점자로 표현하려면 먼저 글자를 초성, 중성, 종성으로 분리하여 각각에 해당하는 점자로 연결하는 방식을 사용한다.

표 1. 한글 점자 표기법  
Table 1. Korean braille notation

초성	중성	종성	약자
ㄱ	ㅏ	ㄷ	가
000100	110001	100000	110101
ㄴ	ㅑ	ㄹ	사
100100	001110	010010	111000
ㄷ	ㅓ	ㄷ	그러나
010100	011100	001010	100000 101110
생략	생략	생략	생략

초성, 중성, 종성이 분리되면 ‘표 1’을 참고하여 점자로 변환한다.[3] 점자는 6점으로 구성되어 있고 좌측에 세로로 구성된 3개의 점이 1,2,3점이고 우측의 세로로 구성된 3개의 점이 4,5,6점으로 불린다.

예를 들어 좌측상단의 하나의 점만 튀어나왔다면 ‘100000’으로 표현할 수 있고, 좌측 가운데 점과 우측 상단의 점이 튀어나왔다면 ‘010100’으로 표현할 수 있다. 이진수로 표현된 점자는 점자폰트와 연결 가능한 약속된 기호로 최종 변환하여 점자폰트를 통해 표현된다.[5, 6]

한글점역은 약자처리를 포함하여 다양한 예외처리 규칙이 사용되고 있다. 약자처리는 ‘가’, ‘사’, ‘은’, ‘을’, ‘온’, ‘언’, ‘영’ 등이 있으며, ‘그리고’, ‘그러나’, ‘그래서’, ‘그런데’, ‘그러면’, ‘그러므로’ 등도 포함된다. 이와 같은 글자는 약자로 표현되어 각 글자마다 초성, 중성, 종성에 해당하는 점자를 찾는 것이 아니라, 해당 약자에 연결된 축약된 별도의 점자를 사용한다. 이와 같이 약자를 사용하는 이유는 점자가 초성, 중성, 종성 형태로 구현하기 때문에 점자의 길이가 상당히 길어짐을 방지하고 시각장애인이 빠른 읽기가 필요하기 때문이다. 그밖에 예외처리로는 초성의 경우는 ‘ㅇ’는 점자에서 생략되어 점자코드로 정의되지 않는다. 또한 ‘ㄱ’, ‘ㄷ’, ‘ㅌ’, ‘ㅍ’, ‘ㅈ’ 등 된소리 초성은 자음을 두 번 표기하지 않고 된소리 지정코드(00001)를 초성에 선행하여 표현한다. 중성의 경우는 ‘ㄱ’, ‘ㄷ’, ‘ㅌ’, ‘ㅍ’는 각각 ‘ㄱ’, ‘ㄷ’, ‘ㅌ’, ‘ㅍ’에 ‘ㅇ’를 추가하여 점자로 표기하는 규칙에 따르기 위하여 별도로 정의되지 않는다. 또한 한 음절을 구성하는 모음 중에서 초성 뒤에 오는 모음 ‘ㅏ’는 생략될 수 있다. 종성은 쌍받침을 표기하기 위해서는 종성코드를 연속적으로 나열한다.[1]

### 2.2 악보 처리를 위한 음악점역

음악점역을 위한 음악점자관련 교재로는 한국시각장애인복지관에서 나온 음악점자교재가 있다. 교재를 보면 음표, 쉼표, 음자리표, 변화표, 조표, 박자표, 슬러 등 23개의 목차로 구성되어 있다. ‘표 2’처럼 각 항목에 해당하는 점형이 약속되어 있다.[4]

표 2. 음표의 점자 표기법  
Table 2. Score's braille notation

음표구분	도	레	미	..
8분/128분	100110	100010	110100	..
2분/32분	101110	101010	111100	..
4분/64분	100111	100011	110101	..

음악점자 변환도 한글점자 변환처럼 다양한 예외 규칙이 있다. 먼저 마디수를 표기하고 악보를 시작하며, 음표가 시작될 때 ‘길’ 표시를 해준다. ‘길’ 표

시는 우리가 알고 있는 옥타브 개념으로 같은 ‘도’라도 어떤 옥타브의 ‘도’인지를 확인하게 해 준다. ‘길’ 표시는 같은 길의 음계가 연속적으로 나올 때는 계속표기하지 않으며, 길이 바뀔 때만 표기를 해 준다. ‘분별기호’는 동일표현의 다른 음표가 연속으로 나올 경우 사용한다. ‘표 2. 음표의 점자 표기법’에서 보면 온음표와 16분음표가 동일한 점자로 표현된다. 이때 동일표현의 음표가 연속적으로 나올 경우 어떤 음표가 온음표이고 어떤 음표가 16분음표인지를 구분하기 어렵다. 이 경우 ‘분별기호’를 사용한다. ‘점역자주’는 한 마디가 한 줄로 표현하기 어려울 정도로 길 경우 점역자가 아직 마디가 끝나지 않았다는 표시를 한 후 다음 줄에 이어서 점역을 하는 것을 말한다. ‘숫자표기’는 조표가 4개이상 표현할 경우는 모두 표현하지 않고 숫자로 표시한다. ‘집합음표’도 마찬가지로 짧은 길이 음표들이 연속적으로 나올 경우 묶어서 표현하는 약속이 있다. ‘슬러’도 5개부터는 여는슬러와 닫는 슬러 사이에 악보점자를 표현한다.[4]

### 3. 음악점자 변환기 설계

본 논문에서 제시하는 음악점자 변환기는 비점역자도 점자악보를 작성할 수 있게 하는 것이 핵심 설계사상이다. 그러므로 입력부에서 모든 악보기호의 입력은 마우스를 이용한 선택적 입력 형태를 취한다. 표현부도 점자의 표현과 함께 실제 악보모양을 함께 제공한다.

#### 3.1 구현범위 및 주요구성

음악점자 구현범위는 음악 점역사가 공부하는 음악점자 교재 범위 중 복잡하지 않은 악보를 표현할 수 있는 음표, 쉼표, 길표, 조표, 박자표, 타이, 슬러, 도돌이표 부분을 구현하였다.

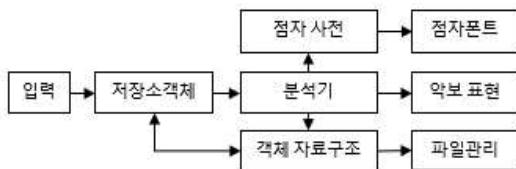


그림 1. 음악점자 변환기 흐름  
Fig. 1. Music-to-braille translator flow

전체적인 음악점자 변환기의 구성은 ‘그림 1’처럼 입력, 저장소객체, 분석기, 객체 자료구조, 점자사전,

점자폰트, 악보표현, 파일관리로 구성된다.

흐름을 살펴보면 비점역자가 입력 부분에서 악보, 쉼표, 조표 등을 선택하면 각각의 객체형태로 저장되어 객체 자료구조에 등록된다. 저장된 객체는 분석기를 통해 한글 형태소 분석 처리와 악보 구성을 위한 예외규칙 처리를 한다. 처리된 내용은 점자사전을 통해 해당 점자로 번역되어 폰트를 거쳐 점자로 표현된다. 악보표현은 점자사전과 관련 없이 자체 알고리즘으로 바로 표현된다. 입력한 상태를 저장하기 위해서는 객체 자료구조를 직렬화하여 파일로 저장하고, 이후 불러오기 기능으로 수정 및 추가 작업을 진행할 수 있다. 입력된 값의 해당 점자를 가져오는 부분의 코드 간결성을 위해 데이터사전기법을 이용한 점자사전을 구성하였으며, 이를 통해 조건문과 반복문을 최소화 하였다. 음악기호 각각의 상태를 2차원 형태의 객체 자료구조로 설계하여 입력 값을 유지하였으며, 표현부와 독립적으로 설계하여 파일포맷 저장 등 다양한 추가기능에 유연성을 극대화 하였다.

#### 3.2 저장소 객체의 구성

음악점역에 필요한 한글, 음표, 쉼표, 길표, 조표, 박자표, 타이, 슬러, 도돌이표는 각각의 객체로 구성한다. 모든 객체는 점자표현을 기본으로 보유하고 있기 때문에 점자표현 객체를 상속받고 있다.

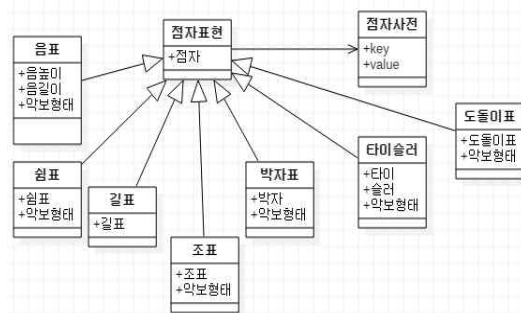


그림 2. 데이터 클래스 구성  
Fig. 2. Entity class diagram

‘그림 2’를 보면 입력된 값은 각각의 객체의 속성 값에 저장된다. 저장된 속성 값을 기반으로 각 객체는 자체적인 처리규칙을 생성자로 가지고 있으며, 점자형태의 이진수로 변환하여 점자사전으로 전달한다. 최종 전달된 점자형태의 이진수 값은 점자사전에 등록된 폰트기호로 변환되어 점자폰트로 전송된다.

### 3.3 분석기 구현

분석기에서는 한글의 형태소분석과 각종 예외처리를 한다. 예외처리가 완료되면 점자사전을 통해 점자폰트에 값을 전달하여 점자를 표현한다. 동시에 악보형태로 보여주기 위해 악보 값으로 변환하여 악보로 표현한다.

```

H_Cho =
    "ㄱ ㅋ ㆁ ㄷ ㅌ ㄹ ㅍ ㅈ ㅊ ㅅ ㅆ ㅇ ㅅ ㅈ ㅊ ㅅ ㅆ ㅇ ㅅ ㅈ ㅊ ㅅ ㅆ";
H_Jung = " ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ";
H_Jong = " ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ";

if (char == 'abbr') then //약자사전
    abbr_Search(); //약자처리
else
    tot = char - 44032;
    cho = tot / (21 * 28);
    H_Cho[cho]; cho_exception(); //초성
    tot = tot % (21 * 28);
    jung = tot / 28;
    H_Jung[jung]; jung_exception(); //중성
    jong = tot % 28
    H_Jong[jong]; jong_exception(); //종성
end if;
    
```

그림 3. 한글 형태소 분석  
Fig. 3. Korean language parser

한글은 유니코드로 44032번째부터 있고, '그림 3'처럼 형태소를 구별한다. 이때 점자는 약자를 가지고 있으므로 약자 데이터사전에 있는지를 먼저 처리한 후 실행한다. 또한 2.1에서 설명한 초성, 중성, 종성에서의 예외처리를 초성, 중성, 종성별로 수행한다.

악보의 예외처리는 길표, 분별기호, 점역자주, 숫자표기, 집합음표, 슬러, 마디내 반복, 마디전체 반복을 처리하여야 한다. '그림 4'에서 보듯이 악보의 마디가 끝날 때마다 객체 자료구조에 저장되어 있는 내용을 검색하여 예외처리 내용이 발견되면 예외처리를 진행한다.

악보의 표현은 GDI(Graphics Device Interface)를 사용하여 '그림 5'형태로 클래스를 구성하여 작성하였다. 최종 화면은 사용자컨트롤로 작성하여 메인프로그램에 삽입하였다.

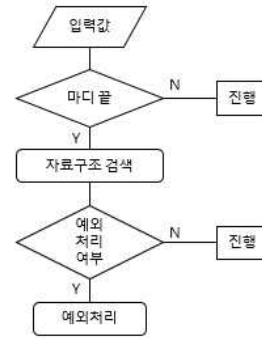


그림 4. 악보 예외처리 알고리즘  
Fig. 4. Score exception algorithm

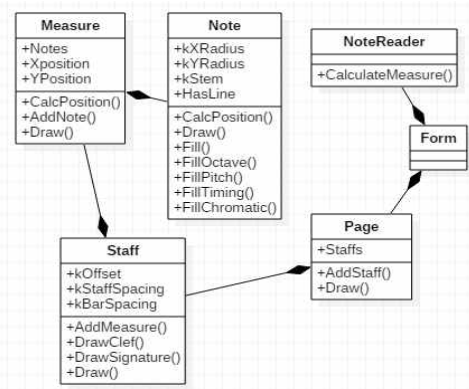


그림 5. 악보표현 구성도  
Fig. 5. Score display layout

### 3.4 객체 자료구조 구성

3.1에서 구성된 저장소 객체는 분석기에서의 예외처리를 편리하게 하고 파일저장 및 검색의 용이성을 위해 객체 데이터사전형태의 자료구조로 동적으로 구성하였다. '그림 6'을 보면 데이터사전을 선언하여 2차원 배열의 키-값 형태로 객체를 저장한다.

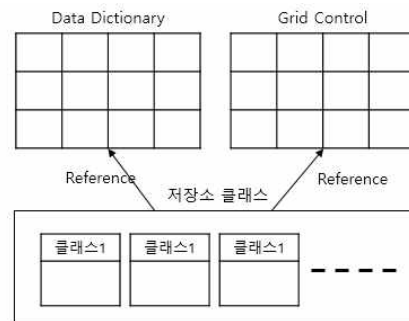


그림 6. 저장소 객체의 데이터사전 구성  
Fig. 6. Entity class data dictionary

이렇게 생성된 객체 자료구조는 그리드 형태의

점자표현부와 동일한 모양을 갖게 설계하였다. 객체 자료구조와 그리드 형태의 점자표현부에는 저장소 객체의 주소 값만을 보유하여 상호 동기화된다. 이처럼 2차원 배열형태의 데이터사전으로 객체 자료 구조를 구현하면 표현부가 바뀌어도 손쉽게 연계가 가능하고 파일 저장 및 예외 처리가 손쉬워진다.

### 3.5 데이터 사전의 구성

한글과 음악기호에 해당하는 점자와 예외처리 일부를 구현하는 점자사전을 데이터 사전방식으로 구현하였다. ‘그림 7’을 보면 입력부에서 ‘8분음표’와 ‘도’가 입력되면 해당 이진수 형태의 점자를 찾을 수 있다. 이진수 형태의 점자는 다시 점자폰트와 약속된 기호로 변환되어 점자로 표현된다. 이와 같은 방법으로 쉼표, 길표, 조표, 박자표 등도 동일하게 구성하였다.

```
class BrailleTable
{
    public static Dictionary<String, Dictionary<String, String>>
        Note = new Dictionary<String, Dictionary<String, String>>
        {
            {"8분음표", new Dictionary<String, String>{{"도", "100110"}, {"레", "100010"},
            {"2분음표", new Dictionary<String, String>{{"도", "101110"}, {"레", "101010"},
            {"4분음표", new Dictionary<String, String>{{"도", "100111"}, {"레", "100011"},
            {"64분음표", new Dictionary<String, String>{{"도", "101111"}, {"레", "101011"},

class Brailldata
{
    public static Dictionary<String, String> chosung = new Dictionary<String, String>
    {
        // 공통점자
        {"000001", "."}, {"000010", "♯"}, {"000011", "♭"},
        {"000100", ""}, {"000101", "."}, {"000110", ""}, {"000111", "."},
        {"001000", ""}, {"001001", "-"}, {"001010", "9"}, {"001011", "0"},
        {"001100", "/"}, {"001101", "*"}, {"001110", ">"}, {"001111", "#"},
```

그림 7. 데이터 사전 구성  
Fig. 7. Data dictionary

한글의 경우는 초성, 중성, 종성, 1종 약자에 대해 데이터사전을 이용하였다. 입력된 한글가사는 먼저 초성, 중성, 종성으로 형태소를 분석한 후 각각의 데이터사전을 통해 해당 점자를 얻어온다. 이때 형태소 분석 전에 약자처리를 우선처리하기 위해서 약자 데이터사전을 먼저 검색한 후 형태소를 분석한다. 또한 된소리 초성처리와 초성 뒤에 오는 모음 ‘ㅏ’의 생략 등 다양한 예외처리도 모두 데이터사전에 등록해서 처리한다.

### 4. 구현 및 평가

음악점자 변환 프로그램 개발을 위한 툴은 마이크로소프트의 Visual Studio 2012를 사용하였고, 사용 언어는 C#, 사용자 인터페이스는 MDI(multiple

document interface)환경으로 구성하였다. ‘그림 8. 화면구성도’를 보면 좌측이 가사 및 음악기호를 선택하는 입력부이고 우측이 표현부이다. 표현부의 상단은 그리드로 구성되어 점자를 표현하고 하단은 실제 악보를 표현한다.

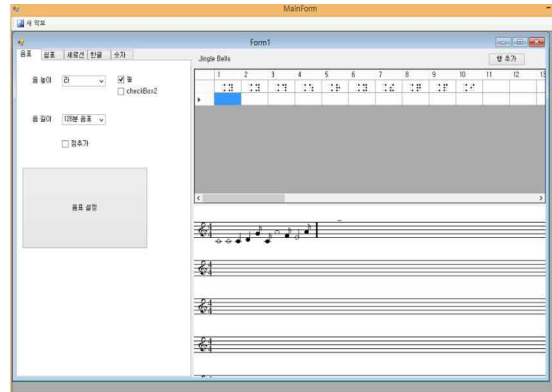


그림 8. 화면 구성도  
Fig. 8. UI layout

작성방법은 먼저 우측에 있는 행추가 버튼을 누르면 그리드에 한 줄이 표현되고 특정 셀을 선택한 후 좌측에 음악기호를 선택하여 적용버튼을 누르면 값이 점자로 변환되어 그리드에 표현된다. 이때 우측 하단부에 있는 악보표현부에 점자의 실제 악보 모양이 표현된다. 수정을 원할 경우는 수정할 셀을 선택하면 입력된 값이 좌측 입력부에 다시 표현되며, 사용자는 다른 기호를 선택하여 적용버튼을 누르면 데이터는 수정된다. 최종 입력된 점자는 ‘파일로 저장’을 선택하여 보관할 수 있으며, 보관된 파일은 다시 불러와서 수정 또는 추가 작업을 진행할 수 있다.

현재 구현된 프로그램은 음악점자 전체를 구현한 것이 아니므로, 이 부분에 대한 확장이 필요해 보인다. 또한 입력 값을 마우스 선택이 아닌 일반 악보를 생성하는 프로그램과 연계하여 작성하는 부분도 고려해 보아야 한다.

### 5. 결론

장애인접근성은 점차 중요성이 강조되고 있다. 점자의 경우는 시각장애인이 다양한 분야의 책을 접할 수 있게 하는 아주 소중한 방식이다.[7, 8, 9] 글자는 점역사 도움 없이 점자 변환프로그램으로 변환이 가능하지만, 음악의 경우는 비점역사가 점자로 변환할 수 있는 툴은 개발되어 있지 않다. 인쇄된 악보를 인식하여 어느 정도 점자로 변환한 후 점역

사가 수정하는 프로그램이 있으나, 악보 인식에 한계가 있고, 결국 점역사가 많은 부분을 처리해야 함으로, 비점역사가 작성할 수는 없다.

본 논문은 점자를 잘 이해하지 못하는 사람도 음악점역을 할 수 있는 ‘음악점자 변환기’를 설계 및 구현 하여 실제 악보가 성공적으로 변환함을 확인하였다. 프로그램의 입력부는 악보기호를 마우스만으로 손쉽게 선택할 수 있게 처리하였으며, 표현부는 점자와 더불어 시각적인 악보로도 표현되게 처리하여 가독성을 높였다. 또한 저장소 객체를 객체자료구조에 저장하여 표현부와 분리하였다. 이를 통해 확장성을 높이고, 검색과 파일저장을 편리하게 하였다. 마지막으로 점자사전을 데이터 사전 기법을 이용하여 조건문과 반복문 코딩을 최소화 하였다.

### REFERENCES

[1] Chung-Han Lee, Kwang-Il Ahn and Sukil Kim, "Design and Implementation of A Korean-To-Braille Translator for A Document Porcessor", Journal of The Research Institute for Computer Science, vol. 2, no. 1, 1994.

[2] Music Braille, <http://www.musicbraille.org/>

[3] Korea Blind Union, <http://www.kbuwel.or.kr/>

[4] Siloam Center For the Blind, [http://www.silwel.or.kr/v2/modules/\\_special/musicbraille.php](http://www.silwel.or.kr/v2/modules/_special/musicbraille.php)

[5] Tae-Yeong Kihl, Sukil Kim and Hong-Gi Kim, "Improvement of korean Braille-Code System for Automatic Reverse Braille Translation " The KIPS Transactionsty, vol. 05 no. 03, 1998.

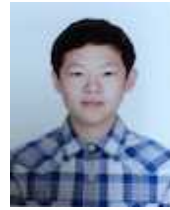
[6] Design of Braille cell Setting Actuators for the Application in the Braille Mouse Concept, Nobels, Korean Institute of Electrical Engineers vol. 4B no. 1, 2004.

[7] K. S. Hong and H. K. Min, "A Study on the Smart Device of Accessibility for Persons with Disabilities", ReHabilitation Engineering And Assistive Technology Society of Korea vol. 9, no. 1, pp. 23-28, 2015.

[8] K. S. Hong and H. K. Min, "The Improvement of Smart Devices Apps Accessibility for Student with Disabilities", ReHabilitation Engineering And Assistive Technology Society of Korea, vol. 8, no. 2, pp. 125-131

[9] Yun-Gun Nam, Hong-Ki Min, "Web Accessibility Implementation using Extensible Markup in the

SVG-based reports", KOREAN SOCIETY FOR INTERNET INFORMATION, vol. 17, no. 1, 2016.



**남윤건(Yoon-kon Nam)**

2015년 - 현재 서울 보인고등학교

Interest: accessibility



**민홍기(Hong-Ki Min)**

1991년 - 현재 인천대학교 교수  
 1993년 - 1994년 Univ. of Delaware 방문교수  
 1985년 - 1991년 한국과학기술연구원 선임연구원  
 1985년 인하대학교 대학원 전자공학(박사)  
 1981년 인하대학교 대학원 전자공학(석사)  
 1979년 인하대학교 전자공학과(학사)

Interest: accessibility, sensor network, HCI