

# 동적 트리 생성/탐색을 통한 Context 기반 연관 음악 탐색 방법

The method of music exploration using dynamic menu trees based on contexts

이광현, 최창규, 조성정, 성영훈, 김연배

삼성종합기술원, Advanced Systems Research LAB

**요약** 모바일 음악 재생 장치의 보급이 활성화 되면서 모바일 환경에서 음악을 즐기고자 하는 욕구가 확대되었다. 모바일 음악 재생 장치의 저장 공간이 확대되어 한번에 많은 곡을 보유하게 되었지만, 실제적으로 많은 곡 중에서 자신이 원하는 곡을 쉽게 찾을 수 있는 방법이 존재하지 않고 있다. 이런 문제들을 해결하기 위해 해당 음악 고유의 Metadata를 활용하여 쉽게 찾고자 하는 경향이 있지만, 현재까지는 단순한 사용자 인터페이스만 제공하고 있다. 각각의 음악은 음악 고유의 Context도 가지게 되며, 사용자의 음악 감상 로그에 의해 생성되는 개인화 Context 및 같은 음악을 즐기는 다른 사용자들에 의해 생성되는 Social Context 등으로 하나의 음악과 관련된 Context가 증가되는 추세에 있다. 이와 같이 추가되는 Context들은 음악을 탐색하는데 있어 효과적인 수단이 제공되어야 하지만 모바일 음악 재생장치에 있는 음악 탐색 프로그램은 새롭게 추가되는 Context에 의한 효과적으로 대응하지 못하고 있다. 이러한 문제점을 해결하기 위해서 모바일 장치에 Embedded Database Engine을 장착하여 동적 트리 생성/탐색을 통한 Context 기반 연관 음악 탐색 방법을 제안하게 되었다. 본 논문에서는 새롭게 고안된 음악 탐색 방법에 대하여 3가지 사항에 대해서 제시하였다. 첫째, 새로운 Context 추가에 대한 동적 메뉴 추가 방법을 제시하였다. 둘째, 실제로 새롭게 추가된 사용자 인터페이스에 대해서 알아보았으며, 마지막으로 제시된 방법이 얼마나 효과적인가를 계산하여 일반화 하였다.

**핵심어:** Music, Dynamic Menu Tree, Music Search

## 1. 서론

기술의 발달에 따라 대용량 휴대용 음악 재생 장치(mp3 player, mp3 휴대폰) 등의 등장으로 많은 음악 파일(저장 용량 30GB의 경우 약 7,500여 곡)을 작은 장치 안에 가지고 다니는 세상으로 바뀌었다. 이러한 기술적 발전으로 인하여 사용자들은 자신의 모바일 휴대 음악 장치에 음악을 바꿔 넣을 필요 없이 한꺼번에 많은 음악을 넣고 다닐 수 있게 되었다. 하지만, 이로 인하여 새로운 문제점들이 야기 되고 있다. 첫째, 노래가 많아 짐에 따라 자신이 원하는 노래를 쉽게 찾는 새로운 방법이 요구되게 되었다. 둘째, 항상 저장된 노래중 듣는 노래만을 감상하는 Long Tail 현상이 발생하게 되었다.[1] 셋째, 음악을 탐색하는데 있어 모바일 휴대용 음악 기기는 텍스트 정보 입력이 불편하여 새로운 검색 방법이 요구된다. 넷째, 사용자들이 실제 찾고자 하는 방식을 적용한 음악 탐색 방법이 존재하지 않고 있다. 다섯째, 네트워크의 발전에 따라 향후 자신의 모바일 장치에 저장한 음악 뿐만 아니라, 음악 스트리밍 서비스에서 선택해서 사용할 시 새로운 사용자 인터페이스가 요청되고 있다. 여섯째, 개인의 음악 선택 방식이 단계별로 이루어 지는 경우가 많음에도 불구하고 이를 적용한 음악 탐색 방법이 없고, 사용자의 고유의 탐색 방법을 제공하지 못하고 있다.

본 논문은 미디어 콘텐츠의 메타 정보를 이용한 미디어 콘텐츠 탐색 장치 및 방법에 관한 것으로 현재 재생중인 미디어 콘텐츠 자체의 메타 데이터 및 향후 확장될 미디어 콘텐츠 관련 메타 데이터에 유연하고 효과적으로 대응할 수 있는 미디어 콘텐츠 탐색 장치 및 방법을 제시하고자 한다.

전체 구성을 위해서는 3가지 단계를 거쳐야 한다. 첫째, 메뉴에 사용될 Context에 대해서 정확한 타입을 지정해야 한다. 둘째, 첫번째 단계에서 정의된 Context를 메뉴 구성에 적당한 위치를 지정한 후 전체 메뉴 구성에 추가하게 된다. 마지막으로 정의된 메뉴 구성에 따라서 실제 음악을 재생하면서 해당되는 메뉴 선택 시 현재 재생곡과 연관된 음악의 리스트를 화면에 출력하고, 사용자의 선택에 의해 연관된 곡으로 이동할 수 있게 된다.

위와 같은 단계별 정의에 의해 사용자 임의로 구성하는 음악의 Meta 정보 기반 메뉴 구성으로 Network Tree 형태의 메뉴[2][3] 선택이 일반적인 음악 탐색 방법에 비해 얼마나 효과적인가에 대해서 알아보하고자 한다. 많은 곡에서 쉽게 음악을 탐색하기 위해서 다른 여러가지 방법들이 행해졌다.[4][5][6]

본 논문의 구성은 Context 기반 새로운 음악 감상 방법을 먼저 제시하고, 실험에 사용된 자세한 구현 방법 등을 제시

한 후, 실험 결과 및 일반화를 시켜 마무리 하고자 한다.

## 2. Context 기반 새로운 음악 감상 방법

### 2.1 기본 Context 에 의한 메뉴 구성 방법

#### 2.1.1 음악 고유의 Meta 정보

음악의 Context 정보를 담고 있는 Meta 정보는 MP3 음악 파일안에 ID3라는 표준 Tag에 정보가 저장되고 있다. 최신 버전에서는 다양한 부가정보를 담기 위해 Version 2.4까지 확장되었지만, 기본 버전인 ID3 Tag Version 1.0의 예를 들었다. 음악의 Context Meta 정보는 곡명, 아티스트, 앨범명, 발매년도, 장르 정보를 추가적으로 갖고 있다. ID3 Tag Version 1.0을 다이어그램으로 표시하면 아래의 그림1과 같이 Star 모양의 구성을 한다고 할 수 있다.

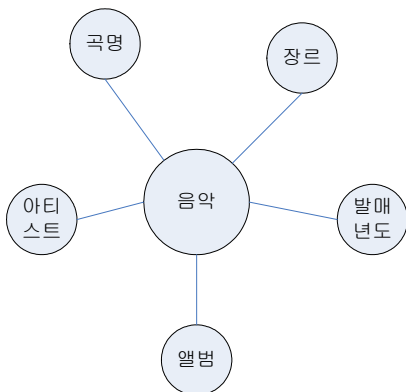


그림 1 ID3 Version 1.0 기반 음악의 Meta data

위와 같은 음악 관련 Meta 정보는 음악 관련 서비스 및 음악 관리를 위해서 다양하게 확장 될 수 있다.

#### 2.1.2 기본 정보를 통한 기존 음악 탐색 방법

기존 음악 감상 방법은 음악 파일의 기본적인 Meta 정보인 곡명, 가수, 앨범명, 발매년도, 장르등의 기본정보를 활용하여 음악을 선택할 수 있는 정보가 제공된다. 음악의 일반적인 구성을 보면, “장르→가수→앨범→곡”등과 같이 메뉴 구성시 계층 구조를 포함하고 있다. 계층 구조 정보를 포함한 새로운 Metadata에 대해서 효과적으로 대응하는 메뉴 트리는 다음과 같은 단계로 이루어 진다. 모바일 음악 재생기의 경우 텍스트 정보의 입력이 불가능하므로 아래와 같은 구조로 음악을 찾게 된다.

- 1단계 : 장르 선택
- 2단계 : 가수(아티스트) 선택
- 3단계 : 앨범 선택 또는 발매년도 선택

위의 1, 2, 3 단계중 일반적인 절차는 단계별로 선택하여 탐색하는 방법을 사용하고 있다. 물론 메뉴의 구성에 따라서 선행 단계를 건너 뛰어 직접적으로 해당 메뉴에 따라 노래를 선택할 수도 있지만 이 방법은 너무나 많은 곡의 리스트가 출력되므로 효과적인 방법이 아니다.

해당 단계에 따라 노래를 선택하는 방법에 대해서는 그림 2에서 도식적으로 설명하고 있다.

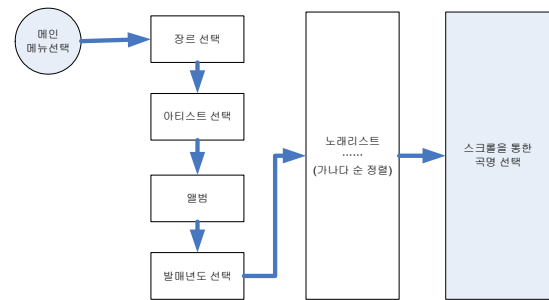


그림 2 새로운 메뉴 추가 단계

### 2.2 동적 Context 확장에 따른 메뉴 기반 음악 탐색

#### 2.2.1 새로운 Context 기반 음악 탐색 방법

새롭게 제안하는 Context 기반 음악 탐색 방법은 현재 재생곡의 확장된 Context에 해당하는 버튼을 누르면, 선택된 Context 메뉴에 해당하는 곡의 리스트를 화면에 출력하여 이중 선택하는 방식을 택하고 있다. 메뉴 구성의 방식은 아래 그림3과 같은 Flow에 의해 선택한 곡을 재생하게 된다.

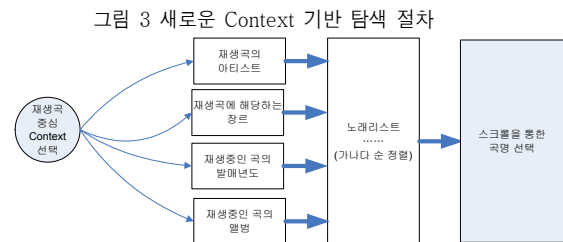


그림 3 새로운 Context 기반 탐색 절차

일반적으로 사람들은 음악을 선곡함에 있어 현재 재생 중인 곡과 관련된 노래를 같이 듣고자 하는 욕구가 있다. 예를 들어 특정 음악을 감상하고 있는데, 같은 시대에 발매된 음악을 듣고 싶은 경우가 있다. 기존의 모바일 음악 재생기를 통해서서는 최상위 메뉴로 접근하여 일일이 해당하는 곡을 찾기 위해 메뉴를 탐색해야 하는 번거로움이 있었다. 음악 파일 재생시 음악 고유의 Meta 데이터 정보를 재생기에서 저장하고 있다가 사용자의 음악 탐색 메뉴 선택에 따라 해당되는 Meta 정보와 메뉴에 정의된 탐색 조건에 의해 바로 장치내의 음악을 탐색하게 된다. 이 방식을 통해 다단계의 탐색이 아닌 간단한 선택에 의해 음악을 쉽게 탐색할 수 있게 된다.

메뉴의 구성 또한 이미 정의된 방식을 사용하는 것이 아니라 사용자의 정의에 따라 미리 Context를 정의하고, 정의한 Context중 사용자의 선택에 따라 메뉴 위치를 정의하여 사용할 수 있는 특징을 가지게 된다. 즉, 사용자의 취향에 맞는 메뉴 구성 방법 또한 제공되게 된다.

## 2.3 Context 기반 동적 메뉴 Tree 생성 방법

동적 메뉴를 구성하기 위한 메뉴의 구성 자체가 사용자 정의에 의해서 이루어져야 한다. 아래 그림4는 실제로 기존에 존재하던 메뉴가 어떤 형태로 이루어지는가에 대해서 도식으로 설명하고 있다. 그림4의 왼쪽 그림에서 유사가수의 메뉴가 추가되는 모습을 보여주고 있다. 오른쪽의 메뉴는 새롭게 추가된 메뉴 Tree의 모습을 나타낸다.

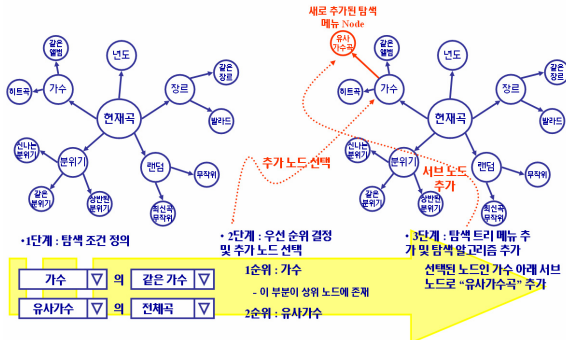


그림 4 Context 기반으로 새로운 메뉴를 추가하는 절차

정확히 위와 같은 단계를 통해 이루어지는 것은 아니지만, 그림으로 메뉴의 구성 자체가 어떻게 변화하는지에 대해서 보여주고자 하였다. 자세한 메뉴 추가는 아래와 같은 3단계로 이루어지게 된다. 첫째, 메뉴의 선택 항목이 될 Context에 대해서 먼저 정의한다. 둘째, 구성된 메뉴 트리에서 추가될 노드를 선택하고, 추가할 Context를 선정한다. 셋째, 추가된 메뉴에 의해 음악 재생시 음악 검색에 사용하게 된다.

### 2.3.1 Context Type 정의

우선 메뉴로 추가되기 위해서는 메뉴 구성을 위한 Context 자체가 어떻게 되는지 정의해야 한다. 예를 들어 장르에 대한 선택 메뉴를 추가하고, 장르 메뉴중 하위 메뉴 구성이 같은 장르, 다른 장르, 유사 장르, 상반된 장르가 있을 경우를 간주하자. 그림5는 이에 상응하는 메뉴 구성 Context Diagram이다.

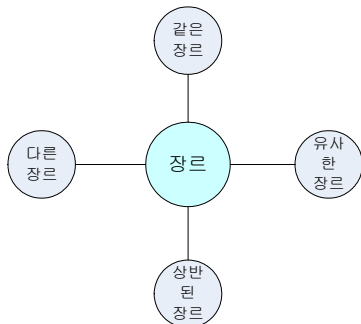


그림 5 장르 메뉴를 정의하는 Context

위의 그림과 같은 메뉴 구성을 위해서 실험용으로 작성한 메뉴 관리 프로그램에서 “Context Type 정의” 버튼을 누르면 아래 그림6과 같이 Context의 이름을 입력하고, 해당하는 Context가 어떤 형태로 분기 메뉴가 구성되어 있는지 체크하면 된다. 아래 그림6과 같이 입력된 Context 정보는 메뉴

에 대해서 관리하는 데이터베이스에 저장되게 된다.

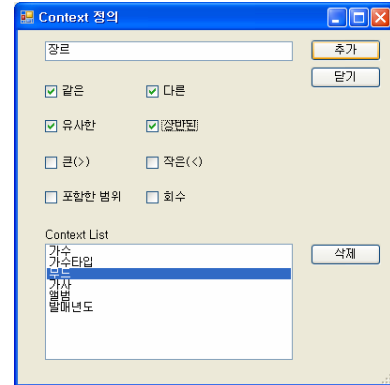


그림 6 장르 메뉴의 Context Type을 지정하는 화면

Context Type을 정의함에 있어 각 노드별로 서비스 메뉴로 분기가 가능한 하나의 레벨로 정의하여야 한다. 2단계 이상의 메뉴를 구성하기 위해서는 각 단계별로 Context Type을 정의하고, 2.3.2에서 정의한 대로 동적 메뉴 트리 정의에 의해 관계를 만들어 줘야 한다.

### 2.3.2 동적 메뉴 트리 정의

메뉴로 추가되기 위해서는 프로그램 메인 메뉴에서 “동적 메뉴 정의” 버튼을 선택하게 되면, 아래 그림7과 같이 기존에 구성되어 있는 메뉴가 표시된다. 상단에서 Tree에서 추가하고자 하는 노드를 선택하고, 하단에 표시되는 Context의 리스트 중 추가하고자 하는 Context 메뉴를 추가한다. “메뉴 추가” 버튼을 누르게 되면 해당되는 메뉴가 추가되게 된다.

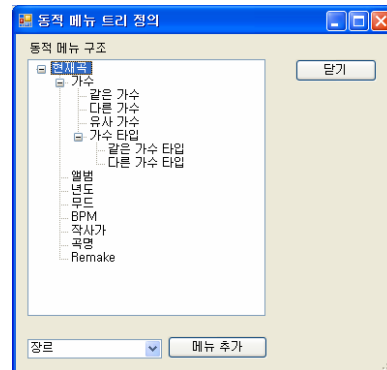


그림 7 장르 Context를 선택하여 최상위 메뉴에 추가하는 화면

같은 Context에 대해서는 다양한 위치에 메뉴 추가가 가능하고, 관리까지 가능하게 된다.

### 2.3.3 동적 메뉴에 의한 음악 선곡

Context Type을 정의하고, 메뉴 구성에 추가된 Context로 실제로 음악을 탐색할 수 있게 된다. 아래 그림 8에서와 같이 현재 재생중인 음악으로부터 관련된 모든 Context를 추출하고, 사용자가 해당하는 메뉴를 탐색하게 되면 현재 음악의 Context와 사용자가 선택한 메뉴와 Context가 일치하는 음악을 전체 음악 파일로부터 검색하게 된다.

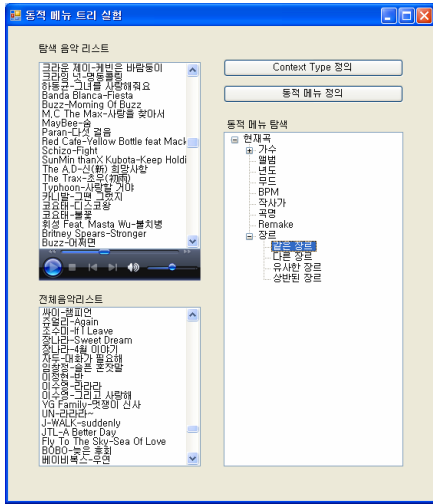


그림 8 새로 추가된 메뉴에 대해서 조회 테스트를 하는 화면

### 3. 실험시스템 구성

#### 3.1 Context 기반 동적 메뉴에 의한 음악 탐색 시스템의 구성

음악 탐색을 효과적으로 구성하고, 동적 메뉴를 구성하기 위해서 시스템 내에 데이터베이스 엔진을 포함하고 있다. 데이터베이스 엔진에서 관리하는 데이터베이스는 2개로써 하나는 메뉴에 대한 관리 역할을 수행하고, 다른 하나는 모바일 음악 재생 장치의 저장소가 관리하는 음악 파일에 대한 정보를 관리하고 있다. 음악 파일 저장소는 실제로 미디어 파일이 저장되어 있는 파일시스템을 의미한다. 시스템의 블록 다이어그램을 보면 아래 그림9와 같다.

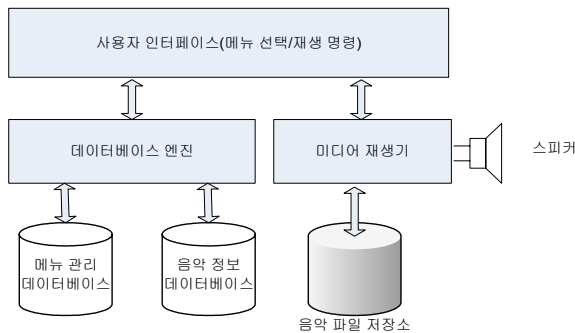


그림 9 전체 시스템 구성도

시스템의 동작은 먼저 현재 재생중인 음악의 모든 Meta 정보를 시스템의 메모리에 관리하고, 메뉴 관리 데이터베이스의 모든 메뉴에 대해서 음악파일이 실제로 존재하는지 여부를 음악 정보 데이터베이스를 통해 확인하고, 보유하고 있는 메뉴에 대해서만 트리 형태로 사용자에게 재생곡 중심의 탐색 메뉴를 제공하게 된다. 사용자가 특정 메뉴를 선택하게 되면, 시스템은 데이터베이스 엔진을 통해 음악 정보를 조회하여 사용자에게 제공하게 되고, 이 리스트 중 사용자의 선택에 따라 음악을 선택, 재생하여 다른 음악으로의 이동이 가능하다.

### 3.2 메뉴 관리 및 음악 정보 관리데이터베이스의 구성

#### 3.2.1 메뉴 관리 방법

음악 탐색에 대한 메뉴를 동적으로 관리하기 위해서는 메뉴 구성에 대한 Meta 정보가 관리 되어야 하는데, 여러가지 방법으로 구성이 가능하지만 쉽게 확장 및 변경이 가능하도록 하기 위하여 데이터베이스를 통해 구현하였다. 아래 그림 10은 메뉴 구성을 관리하기 위한 데이터베이스 스키마이다.

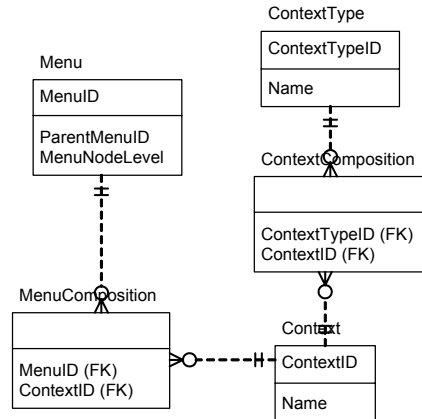


그림 10 동적 메뉴 관리를 위한 데이터베이스 Schema

전체적인 메뉴 구성을 하기 위해서는 우선 Context Type 테이블이 있어야 한다. Context Type은 “같은”, “다른”, “유사한”, “상반된”, “큰”, “작은”, “포함한 범위” 와 “회수”로 이루어 지게 된다. Context 테이블은 사용자의 정의에 따라 Context를 구성하되 추가하는 Context가 어떤 타입인가에 대한 정보를 Context Composition 테이블에 저장하게 된다. 하나의 Context는 여러 개의 Context Type을 포함할 수 있도록 설계되었다. Context의 구성이 완료되게 되면, Context로 메뉴를 구성할 수 있게 되는데, Menu 테이블에는 해당되는 메뉴 이름과 부모 메뉴의 ID가 저장되고, 해당 메뉴가 탐색 메뉴 트리상에서 어느 레벨에 존재하는가를 나타내는 정보를 저장하게 된다. 하나의 메뉴는 하위 여러 개의 Context 메뉴를 포함할 수 있고, 그 정보는 Menu Composition 테이블에 저장되게 된다. 위의 데이터베이스를 통해 음악 탐색에 대한 메뉴 구성이 어떻게 되어 있는가를 식별해 줄 수 있게 된다.

#### 3.2.2 음악 정보 관리 방법

모바일 음악 재생 장치에 보유하고 있는 음악의 정보를 관리하기 위하여 음악 정보 데이터베이스를 유지하여야 한다. 음악 정보 데이터베이스의 역할은 음악자체의 메타 정보 및 위치 정보, 파일정보등을 관리하여 사용자의 메뉴 선택시 해당하는 음악의 보유 여부를 쉽게 조회할 수 있는 기능을 제공하게 된다. 음악 정보 데이터베이스의 구성은 다양한 확장성을 고려하여 설계되어야 하고, 실험에 사용된 데이터베이스는 아래 그림11과 같다.



그림 13 음악을 선곡하는 절차에 대한 예

새롭게 제시된 사용자 인터페이스가 없다면 사용자는 최상위 메뉴에 올라가고, 가수를 선택하며, 가수의 선택에 있어서도 가수를 고르기 위한 기기 조작 회수도 많아지게 되고, 해당되는 곡을 찾기 위한 메뉴를 다 찾아 들어가야 하는 불편함이 따르게 된다.

#### 4.1.3 실제 음악 탐색 응용 프로그램 구현 예

실제로 같은 새롭게 제시된 개념으로 음악 PC에서 재생이 가능한 음악 재생기를 작성하였다. 그림14는 “이효리의 10minutes”라는 노래를 재생 중에 있고, 최초 메뉴로 “분위기 → 다른 분위기”를 선택한 경우를 보여주고 있다. 자신이 현재 음악을 듣고 있으면서 재생되는 음악의 Context와 관련된 다른 음악을 아래 그림과 같이 몇 번의 클릭만으로 선택할 수 있게 해 준다.



그림 14 동적 메뉴 탐색에 대한 실제 응용 프로그램 구성 화면

## 4.2 사용자 조회 회수 비교 Test 및 일반화

실험을 하기 위해서 실제 보유하고 있는 음악 파일에 대해서 실험을 수행하려고 했으나, 실험에 쓰이는 데이터가 일반성이 떨어지게 되어 임의의 가정에 의해 계산에 의해 사용자 조작 회수의 향상에 대해서 알아 보았다.

- 실험 조건 : 1000곡  
(20명의 가수, 가수당 10개 앨범, 앨범당 5곡)
- 제약 조건 : 리스트중 해당 원하는 것을 선택하기 위

해서 동작한 버튼 동작 회수는 생략함.

- 계산 방법 : 평균 메뉴 선택 회수로 계산  
(새롭게 제안된 Music Exploration 방법을 사용하면 모든 메뉴 선택에 대해서 2회만에 찾는다고 가정)

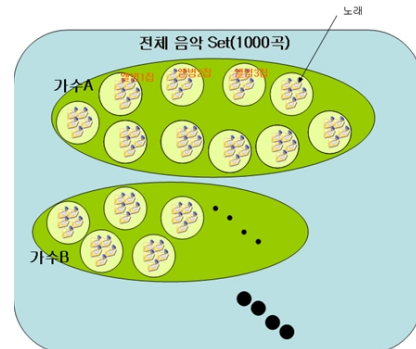


그림 15 실험을 위한 음악 데이터 베이스 구성

- 일반적인 방식에서의 메뉴 동작 회수

- 1회 : 같은 앨범 내 곡
- 3회 : 같은 가수의 다른 앨범 내 곡
- 5회 : 다른 가수의 곡

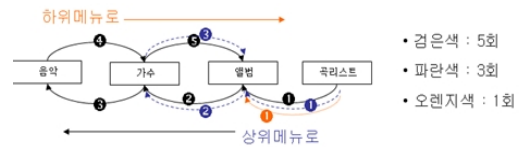


그림 16 음악 종류에 따른 조작 메뉴 절차

$$\text{조작회수} = 1 \times \frac{4}{1000} + 3 \times \frac{45}{1000} + 5 \times \frac{950}{1000} = 4889$$

← 같은 가수의 다른 앨범  
← 다른 가수의 노래  
← 같은 앨범내 곡

그림 17 조작회수 계산 방법

그림 17에서와 같이 간략화 시킨 계산에서도 볼 수 있듯이, 해당되는 곡의 리스트를 보기 위해서 기존의 방식은 평균 4,889회의 메뉴를 동작시켜야 하지만, Music Exploration을 사용하게 되면 평균적으로 2회만으로 사용자가 원하는 감상곡 중심의 노래를 탐색할 수 있다. 일반적인 방법에 대비하여 2배 이상의 조작 회수를 줄일 수 있는 효과를 가져온다.

단순한 조작 회수를 떠나서, 현재 재생되고 있는 음악에 대한 상세한 정보를 알아야지만, 음악을 탐색할 수 있다. 곡 중심의 음악 탐색은 기존에 있는 사용자 인터페이스를 통해서 거의 불가능하다고도 할 수 있다.

위의 식을 일반화 시키기 위해서 곡의 수를 N곡일 경우, 위와 비슷한 조건으로 앨범당 5곡, 가수당 10개의 앨범을 가졌다고 할 경우 메뉴 선택회수(y)에 대한 것을 수식화 시키면 아래와 같다.

$$y = 1 \times \frac{4}{N} + 3 \times \frac{45}{N} + 5 \times \frac{N-50}{N}$$

수식화되어 있는 것에 대해서 그래프를 그려보면 아래 그림 18 및 19와 같다.

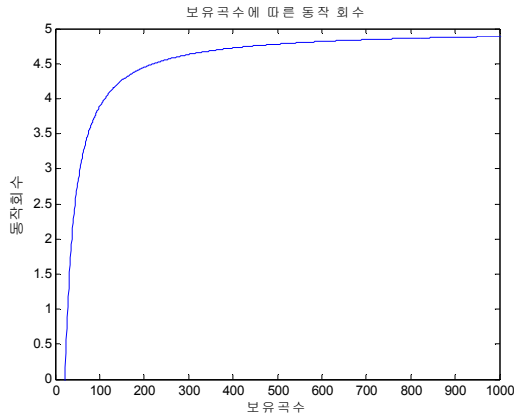


그림 18 음악 보유 곡수에 따른 메뉴 조작 회수(~1000곡)

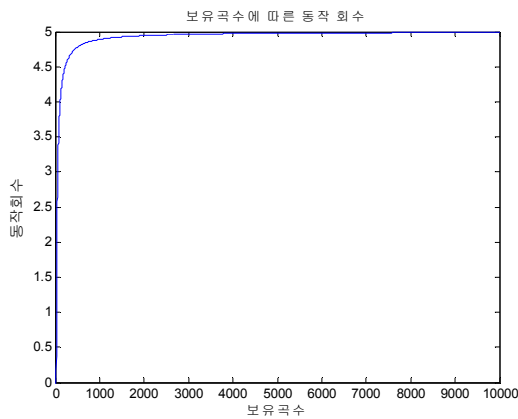


그림 19 음악 보유 곡수에 따른 메뉴 조작 회수(~10000곡)

그림 18 및 19의 결과에서도 볼 수 있듯이 여러가지 제약 조건 하에서 음악 보유 곡수가 늘어날 경우 단순한 메뉴 동작 회수만을 보더라도 보유 곡수가 200곡만 넘어도 평균적으로 5회에 가까운 동작을 통해 음악을 찾을 수 있다. 그 이후에는 동작회수에는 큰 변화를 보이지 않고 있다.

하지만 위의 실험 방법은 보유곡 리스트중 실제 음악을 찾기 위해 위,아래로 이동하며 동작하는 것을 배제한 것이므로, 실제적인 동작회수는 더 많아질 것이다.

기존의 방식에 대비하여 본 논문에서 제시한 방법은 사용자의 동적 메뉴 구성에 따라 적절한 메뉴 동작 회수를 임의로 조정할 수 있고, 사용자 선호도에 따라 메뉴를 구성할 수 있어 효과적인 사용자 인터페이스라고 할 수 있다.

#### 4. 결론

음악 감상에 있어서 개인의 성향에 따라 임의로 모아 놓은 Playlist 재생이 아닌 일반적인 메뉴 탐색을 하되, 재생곡 중심 Context 기반으로 Music Space를 Navigation 할 수

있는 방법이 효과적임이 알 수 있었다. 이는 사용자가 가수 혹은 제목을 기억해야 하는 번거로움이 없이 현재 듣고 있는 음악이 Seed 역할을 하여 해당하는 Context로 사용자가 원하는 형태의 음악 탐색이 가능하였고, 기존의 방식으로 음악을 찾는 방법에 비해 월등히 효과적임을 알 수 있었다. 다른 분야에도 객체간의 관계를 통해 정보를 선택하듯이[7], 음악에 대해서도 같은 시도가 효과적임을 알 수 있었다.

본 실험을 통해 음악에 Context가 추가됨에 따라 자동으로 메뉴의 구성 시 하위 메뉴로 자동으로 생성됨을 확인할 수 있었고, 음악 탐색에 있어서는 더 이상 고정된 메뉴 방식이 아닌 콘텐츠에 포함된 Metadata에 따라 동적으로 메뉴가 구성될 수 있는 방법을 제시하였다. 이를 활용하면 새로운 Meta 정보가 추가되더라도 더 이상 모바일 음악 재생 장치의 펌웨어를 업데이트 할 필요 없이 추가된 Meta 정보를 기반으로 메뉴 구성만 새롭게 갖추어 음악 탐색에 활용할 수 있어 사용자에게 보다 풍부한 탐색 방법을 제공할 수 있게 된다.

이와 같은 방법은 향후에 미디어와 관련된 Context 정보가 추가될 수 있는 모든 미디어 콘텐츠에 적용이 가능한 범용적인 수단이 될 것이다.

#### 참고문헌

- [1] Chris Anderson, "The Long Tail : Why the Future of Business Is Selling Less of More" , 2006
- [2] Heer, J., S.K. Card, J.A. Landay. PREFUSE : Information Visualization Toolkit, CHI 2005, Portland, OR, 421-430
- [3] JUNG : Java Universal Network/Graph Framework. <http://jung.sf.net>
- [4] Rob van Gulik, Fabio Vignoli, Huub van de Wetering, "MAPPING MUSIC IN THE PALM OF YOUR HAND, EXPLORE AND DISCOVER YOUR COLLECTION", ISMIR, 2004
- [5] Elias Pampalk, Andreas Rauber, Dieter Merkl, "Contentbased Organization and Visualization of Music Archives", ACM, 2002
- [6] Marc Torrens, Patrick Hertzog, Josep-Lluís Arcos, "VISUALIZING AND EXPLORING PERSONAL MUSIC LIBRARIES", ISMIR, 2004
- [7] Jeffrey Heer, danah boyd, "Vizster: Visualizing Online Social Networks", 2005