



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0085142
(43) 공개일자 2008년09월23일

(51) Int. Cl.

G06Q 50/00 (2008.03)

(21) 출원번호 10-2008-7014654

(22) 출원일자 2008년06월17일

심사청구일자 2008년07월30일

번역문제출일자 2008년06월17일

(86) 국제출원번호 PCT/US2006/048330

국제출원일자 2006년12월19일

(87) 국제공개번호 WO 2007/075622

국제공개일자 2007년07월05일

(30) 우선권주장

60/752,102 2005년12월19일 미국(US)

(71) 출원인

스트랜즈, 아이엔씨.

미국 오레건 97333 코밸리스 에스더블유 매디슨
760 스위트 106

(72) 발명자

토렌스 마르크

미국 오레곤 97333 코밸리스 에스. 더블유. 매디슨
스트리트 760스위트 106

페레라 페레

스페인 이-08193 바르셀로나 벨라테라 8193

(74) 대리인

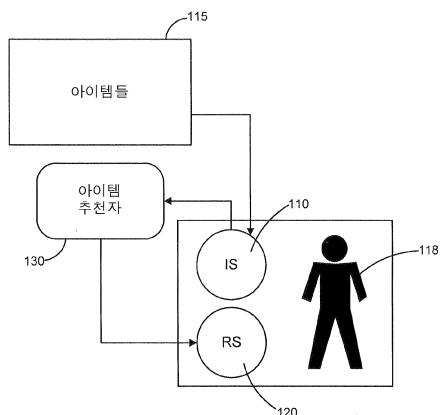
리앤목특허법인

전체 청구항 수 : 총 20 항

(54) 사용자 대 사용자 추천자

(57) 요 약

사용자 커뮤니티 내의 사용자들을 다른 사용자들에게 추천하기 위한 시스템들 및 방법들의 실시 예들이 계시된다. 이런 방법의 일구현에서, 데이터의 2개의 서로 다른 세트들이 고려될 수 있다: a) 사용자가 듣고 있는 (또는 그렇지 않으면 예약하고 있는(engage)) 음악(또는 다른 아이템들), 및 b) 사용자들에게 주어진 음악(또는 다른 아이템들)의 추천들이다. 몇몇 실시 예들에서, 사전계산 방법들이 시스템이 효율적으로 커뮤니티 내의 사용자들 간에서 아이템 세트들 및 추천된 아이템 세트들을 비교하도록 한다. 이런 비교들은 또한 어떤 사용자들이 정해진 타깃 사용자를 위해 추천되어야 하는지를 알아내기 위해 사용될 수 있는 메트릭들을 또한 포함할 수 있다.

대 표 도 - 도1

특허청구의 범위

청구항 1

사용자 커뮤니티 내에서 사용자들을 추천하기 위한 방법으로서,

상기 사용자 커뮤니티 내에서 제1 사용자를 선택하는 단계;

상기 사용자 커뮤니티 내에서 사용자 세트를 선택하는 단계;

상기 제1 사용자를 위한 사용자 프로필(profile)을 상기 사용자 세트 내의 사용자들 각각을 위한 사용자 프로필들과 비교하는 단계; 및

상기 제1 사용자를 위해 추천된 사용자 세트를 발생시키는 단계를 포함하며, 상기 추천된 사용자 세트는 상기 사용자 커뮤니티 내의 적어도 하나의 사용자를 포함하는, 방법.

청구항 2

제1항에 있어서,

상기 추천된 사용자 세트는 상기 사용자 커뮤니티 내의 복수의 사용자들을 포함하는, 방법.

청구항 3

제1항에 있어서,

제1 사용자를 선택하는 상기 단계는 상기 추천된 사용자 세트를 발생시키기 위해 상기 제1 사용자로부터의 요청을 수신하는 단계를 포함하는, 방법.

청구항 4

제1항에 있어서,

상기 사용자 세트는 상기 사용자 커뮤니티 내의 상기 사용자들 각각을 포함하는, 방법.

청구항 5

제1항에 있어서,

상기 사용자 프로필은 복수의 미디어 데이터 아이템들을 포함하는 미디어 아이템 세트를 포함하는, 방법.

청구항 6

제5항에 있어서,

상기 미디어 데이터 아이템들은 복수의 미디어 아이템들을 식별하는 메타데이터를 포함하는, 방법.

청구항 7

제5항에 있어서,

추천된 사용자 세트를 발생시키는 상기 단계는 상기 미디어 아이템 세트 내의 미디어 아이템들에 대한 플레이카운트(playcount)들을 분석하는 단계를 포함하는, 방법.

청구항 8

제5항에 있어서,

상기 사용자 프로필은 복수의 추천된 미디어 데이터 아이템들을 포함하는 추천된 미디어 아이템 세트를 더 포함하는, 방법.

청구항 9

제8항에 있어서, 상기 추천된 사용자 세트는,

상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭(metric), 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 및 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭 중 적어도 하나를 사용하는 것에 의하여 발생되는, 방법.

청구항 10

제9항에 있어서,

상기 추천된 사용자 세트는 상기 사용자 세트 내의 상기 사용자들 각각을 위한 상기 미디어 아이템 세트 내의 미디어 아이템들에 대한 플레이카운트들을 사용하는 것에 의해 또한 발생하는, 방법.

청구항 11

제8항에 있어서,

상기 추천된 사용자 세트는 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 및 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭을 사용하는 것에 의해 발생되는, 방법.

청구항 12

사용자 커뮤니티 내에서 사용자들을 추천하기 위한 방법을 실행하기 위한 컴퓨터 실행가능 명령어들을 저장한 컴퓨터 판독가능 매체로서, 상기 방법은,

상기 사용자 커뮤니티 내에서 제1 사용자를 선택하는 단계;

상기 사용자 커뮤니티 내에서 사용자 세트를 선택하는 단계;

상기 제1 사용자를 위한 사용자 프로필(profile)을 상기 사용자 세트내의 사용자들 각각을 위한 사용자 프로필들과 비교하는 단계; 및

상기 제1 사용자를 위해 추천된 사용자 세트를 발생시키는 단계를 포함하며, 상기 추천된 사용자 세트는 상기 사용자 커뮤니티 내의 적어도 하나의 사용자를 포함하는, 컴퓨터 판독가능 매체.

청구항 13

제12항에 있어서,

상기 제1 사용자를 선택하는 단계는 상기 추천된 사용자 세트를 발생시키기 위해 상기 제1 사용자로부터의 요청을 수신하는 단계를 포함하는, 컴퓨터 판독가능 매체.

청구항 14

제12항에 있어서,

상기 사용자 프로필은 복수의 미디어 데이터 아이템들을 포함하는 미디어 아이템 세트를 포함하는, 컴퓨터 판독 가능 매체.

청구항 15

제14항에 있어서,

상기 미디어 데이터 아이템들은 복수의 미디어 아이템들을 식별하는 메타데이터를 포함하는, 컴퓨터 판독가능

매체.

청구항 16

제14항에 있어서,

추천된 사용자 세트를 발생시키는 상기 단계는 상기 미디어 아이템 세트 내의 미디어 아이템들에 대한 플레이카운트(playcount)들을 분석하는 단계를 포함하는, 컴퓨터 판독가능 매체.

청구항 17

제14항에 있어서,

상기 사용자 프로필은 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트를 더 포함하는, 컴퓨터 판독가능 매체.

청구항 18

제17항에 있어서, 상기 추천된 사용자 세트는,

상기 미디어 아이템 세트와 상기 사용자 세트 내의 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭(metric), 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 및 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭 중 적어도 하나를 사용하는 것에 의하여 발생되는, 컴퓨터 판독가능 매체.

청구항 19

제18항에 있어서,

상기 추천된 사용자 세트는 상기 사용자 세트 내의 상기 사용자들 각각을 위한 상기 미디어 아이템 세트 내의 미디어 아이템들에 대한 플레이카운트들을 사용하는 것에 의해 또한 발생하는, 컴퓨터 판독가능 매체.

청구항 20

제17항에 있어서,

상기 추천된 사용자 세트는 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위한 미디어 데이터 아이템들의 미디어 아이템 세트와의 공통부분을 분석하는 메트릭, 및 상기 추천된 미디어 아이템 세트와 상기 사용자 세트 내의 상기 사용자들 각각을 위해 추천된 미디어 데이터 아이템들의 추천된 미디어 아이템 세트와의 공통부분을 분석하는 메트릭을 사용하는 것에 의해 발생되는, 컴퓨터 판독가능 매체.

명세서

기술분야

<1> 본원에서 설명할 실시 예들과 연계되어 사용될 수 있는 추천자 시스템들의 예들은 "Recommender System for Identifying a New Set of Media Items Responsive to an Input Set of Media Items and Knowledge Base Metrics," 명칭의 미국특허공개번호 제2006-0184558호 및 "Dynamic Identification of a New Set of Media Items Responsive to an Input Mediaset." 명칭의 미국특허공개번호 제2006-0173910호에서 설명된다.

배경기술

<2> 아래의 기술에서, 프로그래밍, 소프트웨어 모듈들, 사용자 선택들, 네트워크 트랜잭션(transaction)들, 데이터베이스 질의들, 데이터베이스 구조들 등의 일정한 특정 세부사항들이 본 발명의 특정 바람직한 실시 예들에 대

한 이해를 통해 제공된다. 그러나 당업자는 실시 예들이 하나 또는 그 이상의 특정 세부사항들 없이, 또는 다른 방법들, 구성요소들, 물질 등과 함께 실행될 수 있다는 것을 인지할 것이다.

<3> 몇몇 경우들에서, 잘 알려진 구조들, 물질들, 또는 오퍼레이션들이 바람직한 실시 예들의 양상들을 불명확하게 하는 것을 회피하기 위해 상세히 설명되거나 도시되지 않았다. 또한, 기술된 특징들, 구조들, 또는 특성들이 다양한 대안적인 실시 예들에서 임의의 적합한 방식으로 결합될 수 있다. 일정 실시 예들에서, 본원에 기술된 방법론들 및 시스템들이 PC들, 랙톱들, PDA들, 및 모든 방식의 다른 데스크탑 또는 휴대용 전자 어플라이언스들에서 일반적으로 찾아지는 마이크로프로세서들의 유형들과 같은, 하나 또는 그 이상의 디지털 프로세서들을 사용하여 실행될 수 있다.

발명의 상세한 설명

<4> 사용자 커뮤니티 내의 사용자들을 다른 사용자들에게 추천하기 위한 시스템들 및 방법들의 실시 예들이 계시된다. 본원에 사용된, "사용자 추천자(user recommender)"는 사용자들의 커뮤니티에 통합된 모듈이고, 이것의 주된 기능은 그 커뮤니티 내의 사용자들을 다른 사용자들에게 추천하는 것이다. 상호 작용할 그 커뮤니티의 사용자들을 위해 그 커뮤니티 내에 아이템들의 세트가 존재할 수 있다. 그 사용자들에게 다른 아이템들을 추천할 아이템 추천자도 또한 존재할 수 있다. 본원에서 설명할 실시 예들과 연계되어 사용될 수 있는 추천자 시스템들의 예들은 "Recommender System for Identifying a New Set of Media Items Responsive to an Input Set of Media Items and Knowledge Base Metrics," 명칭의 미국특허공개번호 제2006-0184558호 및 "Dynamic Identification of a New Set of Media Items Responsive to an Input Mediaset." 명칭의 미국특허공개번호 제2006-0173910호에서 설명된다.

실시 예

<16> 본원에서 설명된 것과 같이, "미디어 데이터 아이템"이라는 용어는 임의의 미디어 아이템 또는 미디어 아이템의 표현(representation)을 포함하는 것을 의미한다. "미디어 아이템"은 노래, 영화, 그림, 전자책, 신문, TV/라디오 프로그램의 세그먼트, 게임 등과 같은, 디지털 미디어 포맷으로 표현될 수 있는, 임의 유형의 미디어 파일을 포함하는 것을 의미한다. 따라서 "미디어 데이터 아이템"이라는 용어는 예를 들어, 재생 가능한 미디어 아이템 파일들(예를 들어, MP3 파일)은 물론, 재생 가능한 미디어 파일(예를 들어 MP3 파일을 식별하는 메타데이터)을 식별하는 메타데이터도 포함하는 것을 의미한다. 따라서 임의의 실시 예에서 "미디어 아이템들"을 사용하는 프로세스, 단계, 또는 시스템을 제공하는 임의의 실시 예에서, 그 프로세스, 단계, 또는 시스템은 대신에 (메타데이터와 같은) 미디어 아이템의 표현, 또는 역으로 사용할 수 있다는 것이 명백하다.

<17> 사용자 추천자는 정해진(given) 타깃 사용자를 위해 관련 사용자들을 선택할 수 있다. 이렇게 하기 위해서, 사용자들은 동등한(comparable) 엔티티(entity)들이어야 한다. 커뮤니티 내의 사용자들을 정의하는 컴포넌트는 사용자 프로필(profile)로 지정될 수 있다. 따라서 사용자 프로필은 2개의 세트들을 정의함에 의해 정의될 수 있어서, 2명의 사용자들을 비교하는 것은 그들의 사용자 프로필들을 공통부분을 찾는(intersecting) 문제가 되도록 할 것이다. 예를 들어, 도 1을 참조하면, 제1 세트는 도 1의 참조 번호 110인 "아이템들의 세트(items set)"일 수 있고, 이것은 특정 사용자(118)를 위해 가장 관련된 아이템들(115)을 포함할 수 있다. 제2 세트는 도 1의 참조 번호 120의 "추천들의 세트(recommendations set)"일 수 있고, 이것은 사용자(118)를 위해 가장 관련된 추천된 아이템들을 포함할 수 있다. 아이템들의 세트(110)는 아이템 사용 및/또는 일정 아이템들과의 일정 사용자의 상호 작용에 의해 추론될 수 있는 반면에, 추천들의 세트는 아이템 추천자(130)를 사용하는 것에 의해 추론될 수 있다. 몇몇 경우들에서, 아이템들의 세트(110)는 추천자(130)의 입력으로서 사용될 수 있기 때문에, 출력으로서 추천들의 세트를 얻을 수 있다.

<18> 아래의 예들 및 구현들에서 서술되는 아이템들은 음악적인(musical) 또는 다른 미디어 아이템들을 위한 것일 것이다. 그러나 본원에 서술된 구현들은 아이템 특정(item-specific)이 아니고 사용자들의 커뮤니티에 의해 사용되고/공유되는 임의의 다른 유형의 아이템으로써 동작할 수 있다는 것이 이해되어야 한다.

<19> 음악적인 또는 멀티미디어 아이템들(트랙들, 아티스트들, 앨범들 등)의 경우, 사용자들은 그것들을 사용하는 것(듣기, 구매하기 등)에 의해 그 아이템들과 상호 작용할 수 있다. 이런 실시 예들에서 그 세트들은 그것들이 음악적인 아이템들을 포함하고 있기 때문에 "음악적인 세트들"로서 지정될 것이다. 따라서 이런 세트들은 "음악 세트(Music Set)" 및 "추천들의 세트(Recommendations Set)"로서 지정될 것이다.

<20> 음악 세트는 사용자가 듣고 있는 아이템들에 의해 형성되는 음악적인 세트이다. 사용자 A의 음악 세트는 본원에

서 Ma로서 표기될 것이다.

<21> 추천들의 세트는 사용자가 추천받고 있는 아이템들에 의해 형성된 음악적인 세트이다. 사용자 A의 추천들의 세트는 본원에서 Ra로 표기될 것이다.

<22> 2개의 사용자 프로필들을 비교하기 위해, 하나 또는 그 이상의 그것들 각각의 세트들의 공통부분(intersection)이 분석될 수 있다. 다양한 서로 다른 메트릭들이 유용한 데이터를 제공하기 위해 공통부분들을 설정하는데 또한 적용될 수 있다. 몇몇 이러한 메트릭들은 사용자들 간의 관계(relation)들을 서술할 것이다. 예를 들어 4개의 기본이 되는 공통부분을 찾는(intersecting) 케이스들은 다음과 같다.

<23> $M_a \cap M_b, M_a \cap R_b, R_a \cap M_b$, and $R_a \cap R_b$. 이런 경우들을 분석함으로써 서로 다른 관계들로서 지칭되거나(labeled) 분류될 수 있는 복잡한 케이스들에 연결될 수 있다. 예를 들어, 일구현에서, 4가지 관련된 관계들이 추출될 수 있다.

<24> 피어(Peer) : M_a 가 M_b 와 공통부분을 충분히 찾으면, B는 A의 "피어"라고 간주된다.

<25> 구루(Guru) : M_a 가 R_b 와 공통부분을 충분히 찾으면, B는 A의 "구루"라고 간주된다.

<26> 피어-구루: 피어 조건 + 구루 조건. B는 A의 "피어-구루"로서 간주된다.

<27> 추종자(Follower) : R_a 가 M_b 와 공통부분을 충분히 찾으면, B는 A의 "추종자"라고 간주된다.

<28> 피어 관계는 그것이 타깃 사용자에게 그의 음악적인(또는 다른 아이템) 라이브러리가 그 타깃 사용자의 음악적인 라이브러리와 일정 방식에서 유사한 다른 하나의 사용자를 제시하기 때문에 관련성이 있을 수 있다. 구루 관계는 그것이 타깃 사용자에게 그의 음악적인 라이브러리가 타깃 사용자가 발견을 좋아할 수 있는 음악을 포함하고 있는 다른 하나의 사용자를 제시하기 때문에 관련성이 있을 수 있다. 피어-구루는 그것이 타깃 사용자에게 피어 관계 및 구루 관계의 친화도(affinity) 및 발견 경험들 모두를 제공하기 때문에 관련성 있을 수 있다. 추종자 관계는 그것이 사용자에게 어떤 사용자들이 그에 의해 영향을 받을 수 있는가를 알 기회를 제공하기 때문에 관련될 수 있다.

<29> 그로부터 전술한 관계들이 예를 들어 추론될 수 있는, 예시적인 구체적인 메트릭들이 이제 개시될 것이다. 메트릭은 입력으로서 2개(또는 그 이상의) 사용자 프로필들을 얻고 출력으로서 측정 가능한 결과를 산출하는 함수(function)일 수 있다. 아래에서 논의되는 메트릭들은, 파라미터들의 순서가 결과를 변경시킬 수 있는 것을 의미하는, 단일 방향이다(unidirectional).

<30> "친화도(Affinity)" 메트릭은 "어느 정도 M_a 가 M_b 와 공통부분을 찾는가(intersect)"의 질문에 답한다. 환언하면, 어느 정도 "친화도 경험"을 사용자 A가 사용자 B에 대해 갖는가?

<31> "발견(Discovery)" 메트릭은 "어느 정도 R_a 가 M_b 와 공통부분을 찾는가"의 질문에 답한다. 환언하면, 어느 정도 "발견 경험"을 사용자 A가 사용자 B에 대해 갖는가?

<32> "지도(Guidance)" 메트릭은 "어느 정도 M_a 가 R_b 와 공통부분을 찾는가"의 질문에 답한다. 환언하면, 어느 정도 사용자 A가 사용자 B를 지도할 수 있는가?

<33> 이런 메트릭들을 써서, 피어 관계들은 친화도 메트릭을 최대화함으로써 찾아질 수 있고, 구루 관계들은 발견 메트릭을 최대화함으로써 찾아질 수 있고, 피어-구루 관계들은 친화도 및 발견 메트릭 모두를 최대화함으로써 찾아질 수 있고, 추종자 관계들은 지도 메트릭을 최대화함으로써 찾아질 수 있다. 하나의 사용자의 다른 하나의 사용자에 대한 전체 관련성이 예를 들어, 메트릭들의 각각(또는 하나 이상)으로써 동작하는 함수를 정의함으로써, 계산될 수 있다. 타깃 사용자를 위해, 커뮤니티 내 모든 다른 사용자들이 X = 친화도, Y = 발견, 및 Z = 지도인, 3차원 공간("ADG 공간")으로의 포인트들로서 위치될 수 있다. [0, 1] 사이의 숫자를 반환하는 것처럼 메트릭들을 정의함으로써, 커뮤니티 내 모든 사용자들이 그 공간 내 1x1x1의 큐브 안에 포함될 수 있다. 도 2는 ADG 공간에서 샘플 타깃 사용자 "X"의 위치를 도시한다.

<34> 설명된 개념 모델을 따르는 사용자 추천자를 구현하기 위해 기본적 시스템이 구축될 수 있다. 하나의 시스템은 아래와 같이 구성될 수 있다.

<35> 1. 사용자 커뮤니티 및 그 커뮤니티 내 각각의 사용자를 위한 사용자 프로필이 추출될 수 있는 아이템 추천자가

존재할 수 있다. 이런 정보는 하나 또는 그 이상의 데이터 소스들에 의해 공급될 수 있다.

<36> 2. 데이터 및 모델의 오퍼레이션들을 구축하고, 데이터 소스들로부터 데이터를 모으는 사용자 추천자의 구현이 존재한다.

<37> 일 구현에 따른 사용자 추천자에 대한 기본 아키텍처 스키마가 도 3에 구현된다. 도면에서 도시된 것처럼 데이터 소스(310)가 데이터를 사용자 커뮤니티(320)로 (그리고 사용자 커뮤니티(320)로부터) 제공한다. 데이터 소스들(310)이 사용자 추천자(330)에 데이터 공급을 또한 제공한다. 사용자 추천자(330)는 사용자 커뮤니티(320)와 상호 작용한다. 특히, 사용자 추천들에 대한 요청들이 사용자 커뮤니티(320)로부터 받아질 수 있고 차례로 추천된 사용자들이 사용자 커뮤니티(320)에 제공된다. 물론, 몇몇 구현들은 추천된 사용자 세트들을 발생시키기 위해 사용자들로부터의 요청들을 수신하는 것에 의존할 수 있고, 다른 구현들은 이런 세트들을 발생시키고 우선적으로 사용자 요청들을 수신함이 없이 그 세트들을 커뮤니티 내의 사용자들에게 제공할 수 있는 것이 계획된다.

<38> 가변(scalable) 아키텍처 해결책을 제공하는 것이 또한 바람직할 수 있다. 요청이 주어질 때, 타깃 사용자를 커뮤니티 내 모든 사용자들과 비교하는 것이 적합하지 않을 수 있다(응답 시간이 사용자들의 수와 비례하여 증가할 수 있다). 이런 문제들에 대한 수많은 해결책들이 구현될 수 있다. 예를 들면, 다음과 같다:

<39> 1. 사용자 데이터가 클러스터화되고(clusterize) 타깃 사용자가 적당한 클러스터(right cluster)와 비교된다.

<40> 2. 사용자들의 고정된 수 또는 서브셋이 사용자 커뮤니티로부터 선택될 수 있다. 사용자들의 이런 서브세트는 "추천 가능한 사용자들(Recommendable Users)"로서 지정될 수 있고 타깃 사용자(들)는 그 크기 고정된(size-fixed) 세트와 비교될 수 있다. 추천 가능한 사용자들은 시스템이 그 커뮤니티 내 가장 흥미있는 사용자들을 추천하도록 하는 일정 절차에 의해 선택될 수 있다.

<41> 음악적인 세트 엔티티는 N이 우리 분야(universe) 내 음악적인 아이템들의 전체 수인 N-차원(N-dimensional) 공간의 희소 벡터(sparse vector)로서 설계될 수 있다. 각 차원은 서로 다른 아이템들을 나타내고, 각각의 구체적 값은 그 아이템의 관련성(relevance)을 나타낸다. 각 아이템에 대해 관련성 값을 더하는 것은 기본적인 시스템 또는 구현이 일정 사용자를 위해 가장 관련된 아이템들을 인식할 수 있도록 한다.

<42> 일정 구현들에서, 아이템들이 음악 트랙들일 수 있다. 그러나 이런 실시예들에서, 2개의 사용자 세트들 내의 아이템들 간의 공통부분들이 (희소(sparsity) 문제 때문에) 덜 예상될 수 있다. 또한, 이런 공통부분들이 계산적으로 비쌀 수 있다.

<43> 이런 문제들이 그 대신에 아티스트 레벨에서 작업하는 것에 의해 일정 실시 예들에서 처리될 수 있다(address). 트랙들 대신에 아티스트들의 공통부분이 있을 가능성이 더 높다. 반면에, 관련성 값이 아이템들이 추출되는 데이터 소스에 의존할 수 있다. 그 결과 표준화(normalization) 처리가 사용되어서, 모든 관련성 값이 질적인(qualitative) 값 규모(scale)과 같이, 알려진 값 규모에 속하게 될 수 있다.

<44> 예를 들어, 도 4는 특정 사용자들에 대한 아티스트들과 같은 특정 아이템들의 관련성을 표시하는 질적 규모(scale)를 도시한다. 정해진 사용자를 위해 C1 보다 작은 관련성 값을 갖는 아티스트들(또는 다른 아이템들)은 "낮음"으로 간주되고, C1 과 C2 사이의 관련성 값을 갖는 아티스트들(또는 다른 아이템들)은 "중간"으로 간주되고, C2와 C3 사이의 관련성 값을 갖는 아티스트들(또는 다른 아이템들)은 "높음"으로 간주된다.

<45> 예시적인 표준화 처리의 세부사항들 및 예들이 일정 사용자를 위한 일정 아이템의 관련성을 찾기 위한 다른 접근 방식과 함께, 이후에 논의된다.

<46> 사용자 엔티티는 본원에서 논의되는 개념 모델에 따라 공통부분들을 계산하기 위해 요구되는 모든 데이터를 갖도록, 고유 ID를 갖는 엔티티 및 2개의 음악적인 세트 엔티티들로서 설계될 수 있다.

<47> 본 발명의 일정 실시 예들에서 구현될 수 있는 일정 오퍼레이션들이 이제 논의될 것이다. 초기 오퍼레이션들은 2개의 사용자 엔티티들을 비교하기 위해 요구되고, 음악적인 세트들 사이의 공통부분들을 포함하고 있는 것들이다.

<48> 예를 들어,

<49> 1. 음악적인 세트의 크기는 수학식 1과 같이 표현될 수 있다:

수학식 1

$$|M_u \cap M_{u'}| = \sum_{k=1}^M \min(M_{uk}, M_{u'k})$$

<50>

<51> 여기서, M_{uk} 는 세트 M_u 내의 아이템 k의 관련성 값이다.

<52> 2. 공통부분의 크기는 수학식 2와 같이 표현될 수 있다.

수학식 2

$$|M_u \cap M_{u'}| = \sum_{k=1}^M \min(M_{uk}, M_{u'k})$$

<53>

<54> 모든 이런 M 아이템들은 M_u 및 $M_{u'}$ 에서 공통된다.

<55> 3. 친화도(Affinity), 발견(Discovery), 및 지도(Guidance) 메트릭들이 아래와 같이 표현될 수 있다.

<56> 친화도 메트릭에 대한 하나의 접근 방식은 수학식 3과 같이 M_u 와 $M_{u'}$ 의 크기를 계산하고 그 크기를 M_u 의 크기에 의해 표준화하는 것으로 구성된다.

수학식 3

$$\text{Affinity}(U, U') = \frac{|M_u \cap M'_{u'}|}{|M_u|}$$

<57>

<58> 다른 하나의 가능한 접근 방식은, R_u 와 $R_{u'}$ 의 공통부분이 어떻게든 친화도 측정(measure)이라는 것을 참작한다면, 이런 인자(factor)를 전체 공식에 추가할 수 있어서 그것을 K 인자로서 가중하는 것에 의해 그 측정을 표준화한다.

수학식 4

$$\text{Affinity}(U, U') = \frac{\frac{|M_u \cap M'_{u'}|}{|M_u|} + \frac{|R_u \cap R_{u'}|}{K \cdot |R_u|}}{1 + \frac{1}{K}}$$

<59>

<60> U' 에 대한 U 의 높은 친화도이 U 에 대한 U' 의 높은 친화도를 반드시 의미하는 것이 아님을 주목해라.

<61>

발견 및 지도에 대한 대응하는 공식이 수학식 5와 같다.

수학식 5

$$\text{Discovery}(U, U') = \frac{|R_u \cap M'_{u'}|}{|R_u|}$$

$$\text{Guidance}(U, U') = \frac{|M_u \cap R'_{u'}|}{|R'_{u'}|}$$

<62>

<63> $\text{Discovery}(U, U') = \text{Guidance}(U', U)$ 이 항상 옳다는 것을 주목하라.

<64>

아래의 모델 오퍼레이션들은 초기 오퍼레이션들을 사용하는 것에 의하여, 사용자 추천자로 하여금 원하는 결과를 계산할 수 있도록 하는 그 사용자 추천자 내에서 구현될 수 있는 실례가 되는 전체적인(global) 오퍼레이션들이다.

<65>

- getBestUsers(사용자, 요청):

- <66> 메트릭들의 일정 세트를 계산함으로써, 추천된 사용자들의 세트가 타깃 사용자를 위해 반환될 수 있다. 요청(requirement)은 어떤 종류의 사용자들이 추천될 것이고 어떤 메트릭들이 고려될 것인지를 특정할 수 있다. 이런 기능(function)을 위한 일반적 알고리즘은 아래와 같을 수 있다.
- <67> 1. TU를 타깃 사용자(Target User)로, RUS를 추천 가능한 사용자들의 세트(Recommendable Users Set)로, REQ를 요청 요구(Requirement of the request)로 하기로 한다.
- <68> 2. RUS 내의 각각의 사용자 U를 위해, REQ에 따라, 필수 메트릭들(TU, U)을 계산하고, 그 결과를 RUS의 비교된 사용자 U와 함께 저장한다.
- <69> 3. 목록의 처음에서 REQ에 따른 최적 사용자들을 갖도록 그 비교 결과에 의해 RUS를 분류(sort)한다.
- <70> 4. RUS의 부목록으로 복귀해서, 처음에서 시작한다.
- <71> - getRelevance(사용자1, 사용자2):
- <72> 사용자2에 대한 사용자1의 모든 메트릭들을 계산함으로써, 부동(floating) 수는 사용자 1에 대해 사용자2가 얼마나 관련성 있는가와 같은 질문에 답하는, 모든 메트릭 값들을 써서 일정 계산들을 실행하는 함수를 계산하는 것에 의해 반환될 수 있다. 이 함수는 예를 들어 ADG 공간 내의 벡터의 길이를 계산할 수 있다.
- <73> 사용자 추천자는 자바 웹 모듈(Java Web Module)로서 구현될 수 있다. 이 모듈은 예를 들어 톰캣(Tomcat) 환경 내 웹애플리케이션(webapp)으로서, 전개될 수 있다. 일정 구현에서, 이런 구현을 위한 데이터 소스들은 다음과 같을 수 있다.
- <74> 1. "도달도(Reach)" API : 각 사용자에 대해 플레이카운트 데이터를 반환한다. 일정 구현들이 이 데이터로부터 음악적인 세트를 추론할 수 있다.
- <75> 2. "UMA" 추천자: 아이템들의 세트에 대해 추천된 아이템들을 반환한다. 몇몇 구현들이 임력으로서 예를 들어, 사용자 프로필의 음악 세트를 사용해서 이런 데이터로부터 음악 세트를 추론할 수 있고 결과적으로 위에서 설명된 사용자 프로필의 추천들의 세트를 얻는다.
- <76> 3. "킬빌(KillBill)" API : 예를 들어, 그것의 별칭(alias)과 같은, 사용자에 대한 일정 여분의 정보를 반환한다.
- <77> 하나의 가변가능한 해결책은 도달도 API를 통해 커뮤니티로부터 최적 N명의 사용자들을 얻고 그들을 추천 가능한 사용자들로 만드는 것이다. 일정 구현들에서, CLUTO 클러스터링(clustering) 프로그램이 사용될 수 있다.
CLUTO 프로그램들은 공개(open source) 소프트웨어이고,
<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download>.에서 다운로드 가능하다. 다른 구현들에서, WEKA 클러스터링 프로그램이 사용될 수 있다. WEKA 또한 공개 프로그램이고
<http://www.cs.waikato.ac.nz/ml/weka/>에서 다운로드할 수 있다. 파일 캐시(file-cache) 프로그램 또한 몇몇 구현들에서 구축될 수 있고 사용될 수 있다.
- <78> 따라서 추천들이 모든 사용자들에게 제공될 수 있더라도, 최적 사용자들만이 추천될 수 있다. 사용자가 최적 N 개의 세트 내에 있지 않다면, 사용자 추천자는 실시간으로 데이터 소스들에 추가될 그 사용자의 사용자 프로필을 요구한다.
- <79> 자바 구현들이 아래와 같이 분할될 수 있는 로직인 클래스들의 세트로서 구성될 수 있다.
- <80> 서브릿 뷰(Servlet View) : 여기에 모든 서브릿 요청/응답 로직이 포함된다.
- <81> 추천자 뷰: 여기에 주된 오퍼레이션들이 종종 그것들을 매니저에게 위임하는, 단독개체.singleton 추천자에 의해 실행된다.
- <82> 매니저 뷰 : 여기서 모든 오퍼레이션들이 실제로 코어 클래스들을 사용하는 것에 의해 실행된다.
- <83> 코어 뷰: 여기서 모델의 토대(foundation)들이 자바 클래스들의 세트에 의해 설립된다.
- <84> 전술한 "뷰들"이 이제 더 자세하게 설명될 것이다. 사용자 추천자는 HTTP 서브릿들의 세트로서 구현될 수 있다. 추천자는 도 5에서 도시되는 것과 같은, 3개의 서로 다른 서브릿들에 의해 사용될 수 있는, 단독개체 인스턴스(Singleton Instance)로서 구현될 수 있다.

- <85> 1. 디버그 서브릿(500)은 디버그 명령들을 따른다. 예를 들어, 실행(action) = 통계(stats)이면 추천자(530)의 내부 메모리 통계들의 그래프를 그리는 것을 실행할 수 있다.
- <86> 2. 추천자 서브릿(510)은 이전에 설명된 모델 오퍼레이션들에 따른 추천 요청들에 따른다.
- <87> 3. 업데이트 서브릿(520)은 사용자 데이터 상에서 업데이트 처리를 실행한다.
- <88> 추천자는 매니저 및 요청 캐시에 의해 주로 형성되는 단독개체 클래스일 수 있다. 캐시로의 동시 액세스들은 일치하지 않은 정보를 막기 위해 세마포어(semaphore)들에 의해 또한 제어될 수 있다. 예를 들어, 디버그 서브릿이 플러시(flush) 캐시 명령을 송신하고, 캐시로부터 결과를 얻기를 기다리는 추천 요청이 존재하는 경우, 그 요청이 그 플러시가 실행된 이후에 처리된다면 널(null) 응답이 수신될 수 있다.
- <89> 몇몇 구현들에서, 그 요청 캐시에 액세스하기 위한 일반적인 알고리즘은 아래와 같다.
- <90> 1. 세마포어를 그것이 열려있다면 닫고; 그렇지 않다면 기다려라.
- <91> 2. 그 결과가 그 캐시 내에 있는가?
- <92> 3. 만약 그렇다면 그 캐시로부터 그 결과를 얻는다.
- <93> 4. 그 캐시 세마포어를 연다.
- <94> 캐시가 해시맵(Hash Map)에서 구현될 수 있고, 그 키(keys)들은 스트링 요청 해시들이고 그 값들은 요청 결과이다. 스트링 요청 해시들이 서로 다른 결과들을 산출할 수 있는 파라미터들의 각각의 세트를 위한 고유 스트링을 얻기 위해 계산될 수 있다.
- <95> 도 6은 사용자 대 사용자 추천자(600) 및 그것의 시스템의 다른 요소들과의 기본적인 상호작용들을 도시한다. 추천자(600)는 매니저(610) 및 요청 캐시(620)를 구비한다. 매니저(610)는 참조 번호 612에서 도시된 하나 또는 그 이상의 외부 연결들을 갖고 참조 번호 614에서 도시된 데이터 공급을 구비한다.
- <96> 매니저는 도 7의 참조 번호 700에 도시된, 서비스 계층 및 데이터스페이스 계층과 같은 2개의 계층들을 갖는 단독개체 클래스로서 구현될 수 있다.
- <97> 서비스 계층(710)은 서비스들(712)의 단독개체 인스턴스(instance)들을 포함할 수 있다. 각각의 서비스(712)는 특정 유형 업무들의 세트를 실행하기 위해 사용될 수 있다. 예를 들어, 데이터에 대해 업데이트를 실행하기 위한 모든 로직을 포함하는, "업데이트 서비스"가 제공될 수 있다. 데이터 간에서 비교기들을 실행하기 위한 모든 로직을 포함하는 "비교기 서비스"가 또한 제공될 수 있다. 이런 서비스들이 데이터스페이스 계층(720)과 직접적으로 통신할 수 있다.
- <98> 데이터스페이스 계층(720)이 외부 서비스들(예를 들어, UMA(722), 킬빌(724), 및 도달도(726))로의 모든 연결들은 물론 추천 가능한 사용자들이 영구적으로 저장되는 메인 메모리 구조들(728)까지도 포함할 수 있다.
- <99> 몇몇 구현들을 위한 기본 클래스들은 도 8에 도시된, 사용자(810), 측정된사용자(MeasuredUser)(820), 사용자유사성(UserSimilarity)(830), 및 요청(840)이다. 이런 구현들에서, 사용자(810)를 위한 음악적인세트(850)가 쌍들(아이템(854) 및 관련성(856))의 한 세트로서 구현될 수 있다. 아이템 및 관련성의 일정 구현들이 (아티스트 ID's를 포함하는) 아티스트아이템(ArtistItem) 및 단순한관련성(SimpleRelevance)(1과 3 사이의 크기)이다. 그 업무들이 입력으로서 그 구현의 이름을 얻어서, 아이템 및 관련성 객체들을 생성하는 것인, 아이템공장(ItemFactory) 및 관련성공장(RelevanceFactory)이 또한 존재할 수 있다. 이런 실시 예들에서 아이템 및 관련성 구현이 다른 코어 클래스들에 영향을 미치지 않고 프로젝트의 어떤 단계에서도 쉽게 변경될 수 있다.
- <100> 사용자가 다른 하나의 사용자와 비교될 때, 사용자유사성(830) 인스턴스들과 함께 비교되는 사용자인, 측정된사용자(820)를 갖는다. 사용자유사성(830)은 얼마나 많은 각각의 메트릭(860)이 타깃 사용자와 서로 관련이 있는가를 특정한다. 클라이언트는 또한 서로 다른 메트릭들을 최대화/최소화하기 위해 요청을 특정한다. 친화도, 발견, 및 지도 메트릭들이 각각 참조번호 862, 864, 및 866으로서 도 8에 도시된다.
- <101> 이전에 서술된 것과 같이, 친화도, 발견 및 지도 메트릭들이 구현될 수 있지만, 다른 메트릭들이 인터페이스 메트릭을 확장하는 것에 의해 또한 구현될 수 있다.
- <102> 인터페이스 메트릭은 각각의 메트릭이 그 결과를 파라미터로 나타낼 수 있는(parametrizable), 2명의 사용자들 간의 공통부분을 찾는것(intersection)을 실행해야만 하는지를 특정한다. 인터페이스는 그 메트릭에 따라 2명의

사용자들 간의 관련성을 측정하는 계산 값을 반환하여야만 하는지를 또한 특정한다. 이 값은 또한 파라미터로 나타낼 수 있어서, 각각의 메트릭이 2중(double) 파리미터화될 수 있다. 계산 값은 0과 1 사이에서 또한 표준화될 수 있다. 예를 들어, 새로운 메트릭 "나이 친화도"가 구현될 수 있다. 이런 메트릭은 공통부분으로서 2명의 사용자 나이 차를 포함하는 부호 붙임 정수(signed Integer) 및/또는 다른 사용자에 대한 하나의 사용자의 질적 인 나이차("더 어림", "많이 더 어림", 등)를 나타내는 스트링(String)을 반환할 수 있다. 1이 2명의 사용자가 동갑인 것을 의미하고 0은 2명의 사용자의 나이차가 너무 떨어져서 시스템의 목적들을 위해 고려될 수 없는 것을 의미하도록 표준화된 계산될 수 있다.

<103> 일예에서 웹애플리케이션(webapp)이 부하 분산장치(load balancer)에 의해 제어되는 2개의 생산 기계들에 배치될 수 있다. 추천 가능한 사용자들의 수는 약 1000이었고 시스템은 하루에 500 요청들 이상에 응답할 수 있었다. 스트레스 테스트가 <http://jakarta.apache.org/site/downloads/downloads_jmeter.cgi>에서 다운로드해서 이용 가능할 수 있는 자바 테스크톱 애플리케이션인, Apache Jmeter로서 만들어 질 수 있었다. 이런 테스트에서 몇몇 요청들이 서버에 송신되었고, 응답시간이 동시 요청 수에 비례하여 증가되었다 테스트 결과 수들은 표 1과 같다.

표 1

요청들	초당 요청들	평균 처리 시간	평균 응답 시간
100	1	60ms	290ms
200	2	110ms	370ms
1000	10	120ms	420ms
10000	100	500ms	1800ms

<104> <105> 몇몇 구현들에서, 볼 수 있는 그래프가 GraphPlotter 툴을 사용해서 그려질 수 있다. 도 9는 GraphPlotter 툴로부터 추출될 수 있는 정보의 예를 제공한다. 도 9에 도시된 것과 같이, 사용자 커뮤니티 내의 다양한 사용자들 간의 발견 및/또는 친화도 값들이 그래프로 표현된다.

<106> 특정 구현들의 추가의 세부사항들이 이제 더 상세히 서술될 것이다. 음악 및 추천 세트들 M_a , R_a 및 M_b , R_b 를 각각 갖는 2명의 사용자들 A 및 B를 갖는다고 가정하자, R이 입력 M으로부터 발생된 아이템 추천자의 출력이라면, 몇몇 아이템 추천자들의 경우, 수학식 6은 항상 옳다.

수학식 6

$$M \cap R = \emptyset$$

<107> <108> 이전에 설명된 것과 같이 4개의 가능한 공통부분들은 $M_a \cap M_b$, $M_a \cap R_b$, $R_a \cap M_b$, 및 $R_a \cap R_b$ 이다. 경우들의 전체 수는 12이다:

<109> 피어 관계: $M_a \cap M_b$. A 및 B는 공통 음악적인 기호들을 갖는다.

<110> 피어-브라더(Brother) 관계: $M_a \cap M_b + R_a \cap R_b$. A 및 B는 공통 음악적인 기호들을 가지고 있고 장래에 또한 공통 음악적인 기호들을 가질 수 있다.

<111> 구루-추종자 관계 : $M_a \cap R_b$. B는 A로부터 배울 수 있고(A가 B에게 구루이고 B는 A의 추종자이다)

<112> 숨겨진 피어 관계: $R_a \cap R_b$. A 및 B는 공통 음악적인 기호들을 갖게 발전될 수 있다.

<113> 피어-구루/피어-추종자 관계: $M_a \cap M_b + M_a \cap R_b$. B는 A로부터 배울 수 있지만, B는 이미 A로부터 어떤 것을 배웠다. 이 케이스는 피어의 특별한 케이스 또는 구루-추종자의 특별한 케이스로서 다뤄질 수 있다. 첫 번째로서 다뤄진다면, 이것을 "더 강한" 피어(두 번째 조건은 사용자 B의 기호(taste)의 다음 "상태"가 또한 A와 B 간의 피어 상태라는 것을 보장한다), 두 번째로서 다뤄진다면, 그것은 "더 약한" 그루-추종자 관계로 간주될 수 있다 (추종자는 구루의 음악에서 자신의 음악의 일정 부분을 볼 것이다).

<114> 피어-브라더-구루/피어-브라더-추종자 관계: $M_a \cap M_b + M_a \cap R_b + R_a \cap R_b$. 위와 동일하지만, 그 외에 추천들에

서 공통부분이 있다.

<115> 정적(Static) 구루-추종자 관계: $M_a \cap R_b + R_a \cap R_b$. B는 A로부터 배울 수 있고 A가 다음 상태로 이동할 때 B는 여전히 A로부터 배울 것이다. 이것이 구루-추종자의 더 강한 경우이다.

<116> 교차 트레인들(Crossing-trains) 관계: $M_a \cap R_b + R_a \cap M_b$. B는 A로부터 배우고 A는 B로부터 배운다. 그러나 이런 사용자들의 다음 상태들은 공통부분을 찾으려 하지 않을 것이고, 그래서 이것이 구루-추종자의 예상 밖의 케이스이다(양방향성이기 때문임).

<117> 택시(Taxi) 관계: $M_a \cap R_b + R_a \cap M_b + M_a \cap M_b$. 위와 동일하지만, 그 외에 음악에서 공통부분이 있다.

<118> 미팅-트레인들 관계: $M_a \cap R_b + R_a \cap M_b + R_a \cap R_b$. B는 A로부터 배우고, A는 B로부터 배우고, 그들의 다음 상태는 공통부분을 찾는 것이다. A 또는 B가 다음 상태로 이동하면, 나머지 사람이 그 이동한 사람으로부터 여전히 배울 수 있다. 둘 다 이동하면, 그들은 피어들이 될 것이다. 이것은 양방향 그루-추종자의 가장 강력한 케이스일 수 있다.

<119> 완벽한 연결 관계: $M_a \cap R_b + R_a \cap M_b + M_a \cap M_b + R_a \cap R_b$. 모든 것이 공통부분을 찾는다(intersect)

<120> 아티스트가 특별한 사용자에게 얼마나 관련있는가를 결정/계산하기 위한 방식들이 또한 존재할 수 있다. 예를 들어, 시스템이 사용자를 위해 아티스트의 플레이카운트들을 갖는다면, 데이터는 일정 수의 플레이카운트들을 "낮다"고 간주할 수 있고, 일정 다른 수의 플레이카운트들을 "중간"으로 간주할 수 있고,, 등이 되도록 절대 컷오프 포인트들을 설정하는 것에 의해 표준화될 수 있다.

<121> 대안적으로 시스템이 사용자를 위한 플레이리스트들의 세트를 갖는다면, 아티스트가 그 플레이리스트들에 나타난 횟수가 계산될 수 있다. 그래서 그 방법론은 위에서 설명된 것과 같이 (즉 컷오프(cut-off) 포인트들을 써서) 처리될 수 있다.

<122> 다른 하나의 대안으로써, 시스템이 추천된 세트를 갖는다면, 아티스트가 추천된 리스트에서 차지하고 있는 위치에 기반한 그 아티스트의 관련성이 계산되고 분석에서 사용될 수 있다. 물론 이것은 추천자가 추천된 아티스트들의 순위매김(ranking) 리스트를 제공한다는 것을 가정한다.

<123> 일정 구현들에서 사용자들이 정해진 아티스트를 얼마나 자주 듣는가, 사용자가 그 아티스트로부터의 얼마나 많은 노래들을 그의 프로필에 갖는가, 및/또는 사용자가 정해진 아티스트와 얼마나 잘 아는가 의해, 사용자들이 더 분류될 수 있다. 예를 들어, 사용자가 듣는 각 아티스트를 위해,

<124> 1. F: 듣는 빈도: 및

<125> 2. K: 지식 (이 아티스트로부터 얼마나 많은 노래들을 사용자가 아는가)

<126> F 및 K의 값들이 높음 또는 낮음으로 분류될 수 있다. 그러므로 특별한 아티스트를 듣는 사람들이 표 2와 같이 분류될 수 있다.

표 2

듣는 사람	빈도	지식
A	낮음	낮음
B	낮음	높음
C	높음	낮음
D	높음	높음

<128> 일반적으로 동일한 유형의 듣는 사람들만이 일치할 수 있지만, 이런 분류들을 완전 제곱(perfect square) 상의 포인트들로서 가정한다면(0은 낮음이고 1은 높음), A는 B 및 C에 대해서 거리 1이고 D에 대해서 거리 $\sqrt{2}$ 이다. 유사하게 B는 A 및 D에 대해 거리 1이고, C에 대해 거리 $\sqrt{2}$, 등이다.

<129> 그러나 듣는 빈도는 지식에 상응하지 않는다. 그래서 한 차원이 다른 차원보다 더 커질 수 있고, 이것이 제곱(square)을 K 차원 근처에서 더 커지게 한다.

<130> 이런 접근 방식으로써, A는 C 가까이에서 높고, B 가까이에서 중간이고 D 가까이에서 낮다. 이런 관계들은 도

10에서 그래픽하게 표현된다. 따라서 정해진 사용자 U를 위한 아티스트 A의 관련성은 따라서 수학식 7과 같이 제공된다.

수학식 7

$$Rel(U, A) = (1 + K(U, A))^2 + f(U, A)$$

<131>

<132> 여기서 $K(A) \subseteq [0, 1]$ 은 사용자 U가 아티스트 A에 관해 갖는 지식을 측정하는 함수이고, $f(A) \subseteq [0, 1]$ 는 이 아티스트를 듣는 사용자 U의 상대 빈도를 반환하는 함수이다.

<133>

몇몇 실시 예들에서, K 는 n/N 으로부터 추론할 수 있고 여기서 n 은 사용자가 아는 특정 아티스트의 노래들의 수이고, N 은 이 아티스트의 노래들의 총계이다. 유사하게, F 는 n/P 로부터 추론할 수 있고 여기서 n 은 사용자가 듣는 특정 아티스트로부터의 플레이카운트들의 수이고, P 는 이 사용자에 대한 플레이카운트들의 총계이다. F 는 몇몇 구현들에서 (위에서 서술된) 도달도 API를 통해 계산될 수 있다.

<134>

위의 기술은 본 발명의 바람직한 실시 예들을 포함하는 발명을 전체적으로 개시한다. 추가적인 상세한 설명 없이도, 당업자는 본 발명을 최대한으로 이용하는 전술한 기술을 사용할 수 있다는 것이 신뢰된다. 따라서 본원에 개시된 실시 예들 및 예들이 어떤 방식으로 본 발명의 범위에 대한 제한으로써가 아니라 단지 예로서 해석되도록 개시된다.

<135>

많은 변경들이 본 발명의 주요 원리들을 벗어남이 없이 위에서 기술된 실시 예들의 세부사항들에 만들어 질 수 있다는 것이 당업자에게 명백할 것이다. 따라서 본 발명은 개시된 특정 실시 예들에 제한되지 않고 수정들 및 다른 실시 예들이 첨부된 청구항들의 범위 내에 포함되는 것이 의도된다는 것을 알 것이다.

<136>

따라서 본 발명의 범위는 아래의 청구항들에 의해서만 결정되어야 한다.

도면의 간단한 설명

<5>

도면들은 본 발명의 일정 바람직한 실시 예들을 묘사할 뿐이고 따라서 본 발명의 제한으로서 여겨져서는 안 된다는 전제에서, 바람직한 실시 예들이 첨부된 도면들의 사용을 통해 전문적으로 상세히 기술되고 설명될 것이다.

<6>

도 1은 일실시 예에 따른 사용자 프로필을 위한 기본 컴포넌트들 및 소스들을 도시하는 다이어그램이다.

<7>

도 2는 일실시 예에 따른 ADG 공간 내 타깃 사용자 "X"의 위치를 보여주는 그래프를 도시한다.

<8>

도 3은 일실시 예에 따른 사용자 추천자를 위한 기본 아키텍처 스키마를 보여주는 다이어그램이다.

<9>

도 4는 사용자 커뮤니티 내의 특정 사용자에 대한 특정 아이템들의 관련성을 표시하는 질적 스케일을 도시한다.

<10>

도 5는 사용자 추천자 시스템의 일실시 예의 서브릿 뷰(Servlet View)를 도시하는 다이어그램이다.

<11>

도 6은 사용자 추천자 시스템의 일실시 예의 추천자 뷰를 도시하는 다이어그램이다.

<12>

도 7은 사용자 추천자 시스템의 일실시 예의 매니저 뷰를 도시하는 다이어그램이다.

<13>

도 8은 사용자 추천자 시스템의 일실시 예의 코어(core) UML(Unified Modeling Language) 다이어그램이다.

<14>

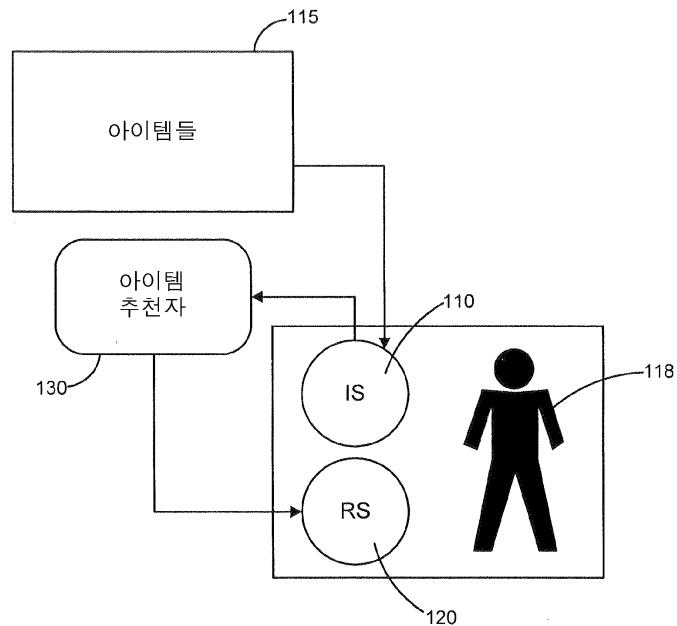
도 9는 사용자 추천자 시스템의 일실시 예에서 사용되는 GraphPlotter 툴로부터 추출될 수 있는 정보의 예를 도시하는 다이어그램이다.

<15>

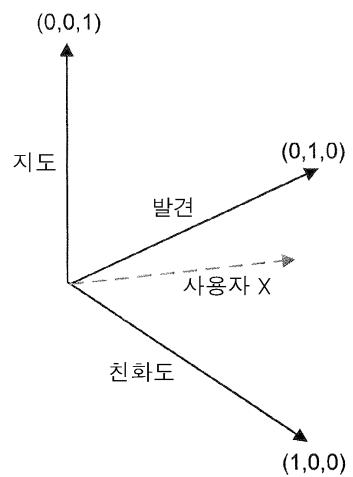
도 10은 "빈도(Frequency)/지식(Knowledge)" 모델에 따른 듣는 사람들의 유형들 간의 관계들을 표현하는 그래프이다.

도면

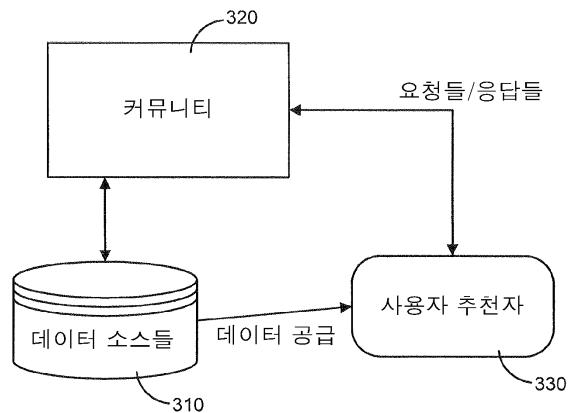
도면1



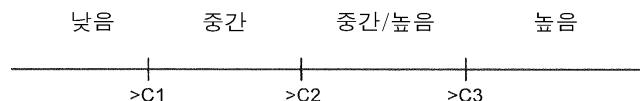
도면2



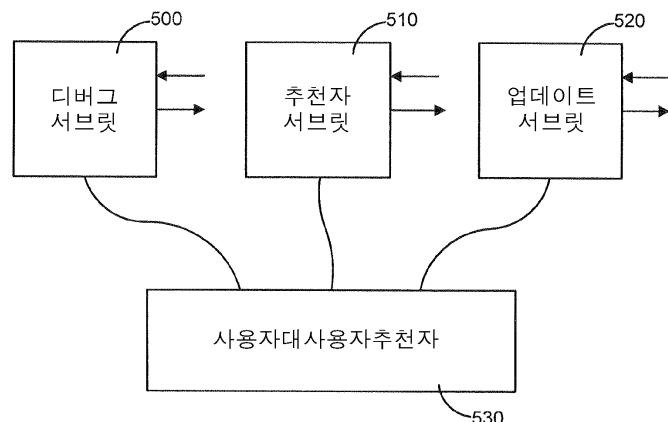
도면3

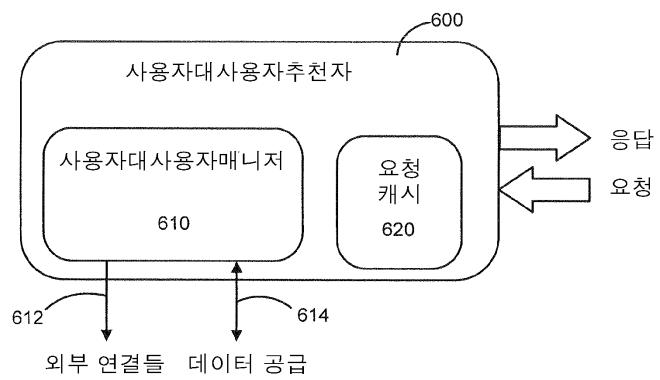
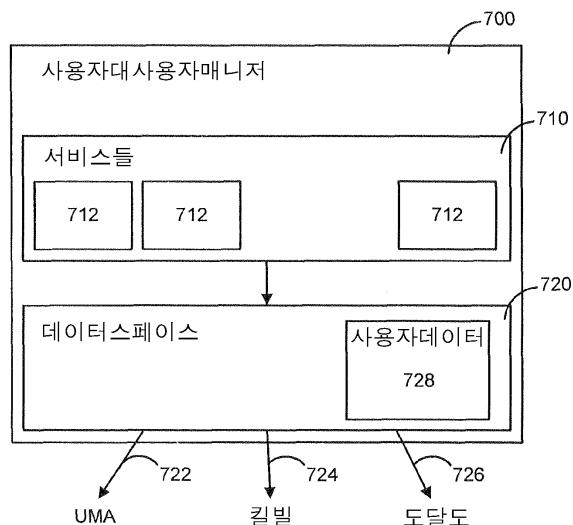


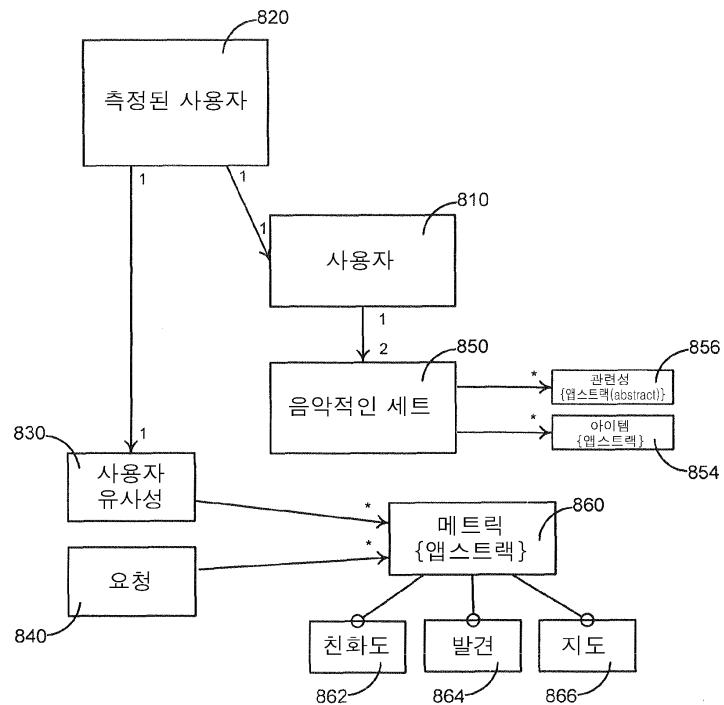
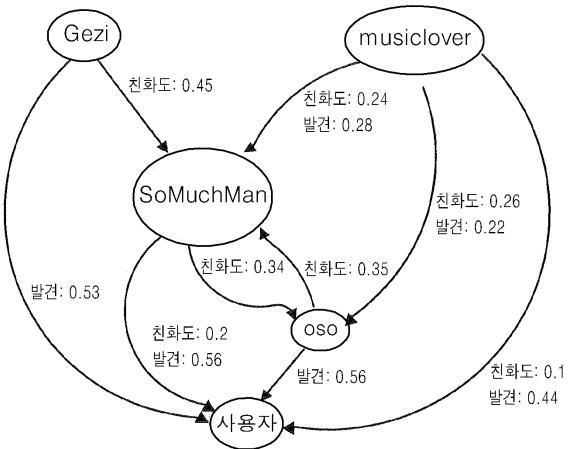
도면4



도면5



도면6**도면7**

도면8**도면9**

도면10

