유비쿼터스 환경에서 다양한 개인화 서비스에 적용하기 위한 사용자 모델링의 일반화 방법론[1)]

이주연° 이성진 이수원
숭실대학교 컴퓨터학과
raniyan@ssu.ac.kr°, ptnrev93@mining.ssu.ac.kr, swlee@ssu.ac.kr

A Generalization Approach to User Modeling for Adapting Various Personalized Services in Ubiquitous Computing Environment

Juyeon Lee° Seongjin Lee Soowon Lee
Computer Science Department, Soongsil University

요 약

최근 연구가 활발히 진행되고 있는 '유비쿼터스'라는 새로운 패러다임은 기존보다 더욱 많은 컴퓨팅 자원을 이용하여 사용자의 편의를 지원하는 것을 그 목표로 하고 있다. 유비쿼터스 컴퓨팅 환경에서 사용자를 지원하기 위한 대표적인 예로 개인화 서비스를 들 수 있으며, 개인화 서비스는 사용자에 대한 모델링이 필수 요소가 된다. 개개인의 행동 패턴 혹은 선호도 정보로 구성된 사용자 모델은 다양한 개인화 서비스의 원활한 지원을 위해 지금까지 유용하게 사용되고 있지만, 기존의 사용자 모델은 각 서비스가 개발될 때, 그 서비스에 적합한 형태로 매번 설계되어야 하는 문제점을 지닌다. 본 논문에서는 이러한 문제점을 해결하고자, 사용자 모델을 구성하는 정보들을 분석하여, 모델 설계에 필요한 일반화된 입력 패턴들을 도출하고, 도출된 패턴들을 바탕으로 더욱 쉽고 빠르게 사용자 모델을 생성할 수 있는 방법을 제안한다.

1. 서 론

나날이 발전하는 컴퓨팅 기술은, 최근 '유비쿼터스(Ubiquitous)'[1]라는 새로운 패러다임을 탄생시켰으며, 다양한 분야에서 연구가 활발히 진행되고 있다. 근본적으로 컴퓨팅 기술은, 인간을 더욱 잘 이해하여, 인간의 삶을 보다 윤택하게 하고 편리하게 생활할 수 있도록 하기위해 발전되어 왔다. 유비쿼터스 컴퓨팅 역시 기존의 많은 기술들을 융합하고[2)], 더욱 많은 자원을 사용하여 사용자의 편의를 지원하는 것이 궁극적인 목표라 하겠다. 특히, 개인화 서비스에 관련된 연구 분야에서 이러한 특징이 두드러지게 나타나며, 개인화 서비스의 효율적인 서비스 관리 및 지원을 위해 '사용자 모델[3)]'을 일반적으로 사용하고 있다.

사용자 모델은 특정 개인화 서비스에 적합한 형태로 설계되어 사용되고 있다. 이에 따라 특정 서비스를 창출하기 위해, 그 서비스에 특화된 사용자 모델을 매번 설계하여 구현하여야 하는 번거로운 문제점을 지닌다. 또한 특정 개인화 서비스에 의존적으로 설계된 사용자 모델은, 서로 다른 개인화 서비스에서 사용자 모델에 대한 정보를 공유하기 어렵기 때문에, 정보의 공유가 필수적인 유비쿼터스 컴퓨팅 환경에서는 적합한 서비스의 형태라 보기 어렵다.

본 논문은 여러 도메인에서의 개인화 서비스를 분석하여, 이에 관한 일반화된 속성을 도출하고, 이러한 일반성에 관한 정보를 바탕으로 빠르고 쉽게 개인화 서비스를 구현할 수 있는 시스템을 제안한다. 또한 일반화된 속성들을 바탕으로 구축된 사용자 모델에 관한 정보는 서로 다른 개인화 서비스에서 공유하기 유리한 장점을 가진다.

유비쿼터스 컴퓨팅 환경에서는 수많은 개인화 서비스가 동적으로 생성되어 서비스 된다. 따라서 빠르고 쉽게 설계 및 구현할 수 있는 개인화 서비스 및 정보 공유를 용이하게 할 수 있는 일반화된 성격을 지닌 사용자 모델은 유비쿼터스 컴퓨팅 환경에 적합한 형태를 취하게 될 것이다.

본 논문은 2절에서 관련 연구, 3절에서 제안한 사용자 모델링 시스템에 대한 설명, 4절에서 실험 및 결과 그리고 5절에서 결론 및 향후 과제의 순서로 구성된다.

2. 관련 연구

추천 시스템에서의 유비쿼터스 사용자 모델링(Ubiquitous User Modeling in Recommender Systems)[2]에서는 사용자 모델에서 사용하는 정보를 공유하기 위해 UMM(User Modeling Mediator)이라는 모듈의 사용을 언급하고 있다. UMM은 여러 개인화 서비스에서 사용하는 사용자 모델에 대한 정보를 일반화된 표현을 사용하여 User Modeling Knowledge Base라는 저장소에 보관하고, 이 정보를 사용하는 개인화 서비스에 적합한 형태로 변환하여 제공하는 일종의 정보 변환기이다. 그러나 개인화 서비스에서 사용하는 사용자 모델의 정보를 어떻게 일반화된 정보로 표현하여, 다른 시스템에서 이용할 수 있게 그 의미 부여를 하는지에 관한 내용은 향후 과제로 남겨두고 있어

1) 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스 컴퓨팅 및 네트워크 원천 기반 기술 개발사업의 지원에 의한 것임
2) IT Convergence
3) 본 논문에서 사용자 모델은 '사용자 선호도 모델' 을 의미한다.

서, 제안한 방법의 적용성 및 효율성에 대한 평가를 하기 어렵다.

Michael Yudelson이 소개한 UMMO(User Modeling Meta - Ontology)[3]에서는 사용자 정보 및 사용자 모델링의 목적, 학습 및 추천에 필요한 알고리즘 등의 많은 사용자 모델링에 관한 정보를 온톨로지로 자세히 기술하여 구축하는 방법을 제안하고 있다. 그러나 유비쿼터스 환경은 앞으로 연구원 및 개발자가 만들어 가야하는 미래의 컴퓨팅 환경이다. 이러한 점을 고려할 때, UMMO에서 필요한 모든 정보를 입력할 수 있게 미리 사용자 모델을 만들어 제시한 방식보다는, 현재 진행 중인 컴퓨팅 환경을 고려하여, 실제 개발자나 도메인 전문가가 에디터를 통해 함축된 사용자 모델을 만들 수 있게 하는 방식으로 일반화를 진행하는 것이 실제 시스템을 구축하고 사용하는데 있어서 더욱 현실적이라 생각한다.

UserML(User Modeling Markup Language)은 사용자 모델을 XML 문서로 표현하고, 또한 사용자 모델 편집 모듈을 사용하는 점에 있어서 본 논문에서 제안하는 사용자 모델링 시스템과 유사한 방법을 사용한다[4]. 그러나 UserML에서는 사용자 모델 편집 모듈을 사용해서, 사용자의 상황 정보 등의 사용자 콘텍스트 정보를 등록하여, 부분적인 사용자 모델을 구성하는 방식을 사용하지만, 본 논문에서 제안한 모델 스키마 편집 모듈에서는 사용자 모델을 구성하는 메타 데이터(meta data)를 설정하는 부분에서 다르다.

3. 사용자 모델링 시스템

현재의 컴퓨팅 기술은 사용자를 더욱 잘 이해하여 보다 좋은 서비스를 제공하는 개인화 서비스에 관련된 분야에 초점이 맞춰지고 있다. 컴퓨팅 개체들과 더불어 사용자의 콘텍스트 정보를 고려한 다양한 서비스가 생성되는 유비쿼터스 환경에서, 다양한 도메인에 적합한 개인화 서비스를 제공하기 위해서는, 사용되는 정보의 일반화가 필요하다. 개인화 서비스에 사용되는 정보를 살펴보자면, 사용자의 선호정보, 학습의 업데이트 주기, 히스토리 및 콘텐츠, 추천 리스트 관한 사항을 들 수 있으며, 이에 관한 자세한 내용은 다음과 같다.

3.1 개인화 서비스에 사용되는 정보들의 일반화

일반적으로 개인화 서비스는 사용자의 서비스 이용 히스토리를 학습하여 사용자 선호도 정보로 구성된 사용자 모델을 구축한다. 구축된 사용자 모델 및 개인화 서비스에 관련된 정보를 구성하는 여러 가지 요소들은 아래와 같다.

- 선호 정보
- 학습 주기
- 히스토리 관리
- 콘텐츠 관리
- 추천 리스트 관리

사용자 정보를 구성하는 요소들은 각각의 개인화 서비스에 적합한 형태로 정제되어 설계 및 구현되어 왔다. 위의 요소들은 기존의 개인화 서비스(예: TV 및 뉴스 추천 서비스)에 특화된 사용자 모델을 분석해서 분류한 것이며, 이렇게 분류된 요소들의 일반화 시킨 과정 및 결과는 다음과 같다.

3.1.1 선호 정보를 나타내는 속성의 일반화

사용자 서비스 이용 히스토리로부터 학습된 각 속성에 대한 선호정보는 사용자 모델을 구성하는 기본적인 요소가 된다. 사용자 모델에 필요한 일반성을 찾기 위해, 선호정보에 대한 속성에는 어떠한 것들이 있는지 기존의 개인화 시스템을 역으로 분석하는 방법을 사용하였다. 예를 들면, TV 추천 서비스에서 선호정보를 나타내는 속성에는 장르, 채널, 방송시간 등이 있으며, 이러한 속성에 대해 일반화를 취하면, 속성 명칭, 타입, 선호도 표현 방식, 속성간의 연관성으로 일반화된 형태를 나타낼 수 있다.

- 속성 명칭 : 장르, 채널, 방송시간 등의 속성명을 나타낸다.

- 속성 타입 : 속성이 가질 수 있는 값에 대한 유형을 나타내며 종류는 다음과 같다.

  · 단일형(Single) : 속성이 한 값을 가지며, 값에 대한 종류 수를 셀 수 없을 경우 사용한다. 예를 들면, TV 추천 서비스에서 프로그램 명(Title)이 이에 해당한다.

  · 열거형(Enumeration) : 속성이 한 값을 가지며, 값에 대한 종류 수가 제한적일 경우 사용한다. 가능한 값들은 따로 명세한다. 예를 들면, TV 추천 서비스에서 채널(channel)이 이에 해당한다.

  · 집합형(Set) : 속성이 여러 값을 가지며, 값의 개수를 지정할 수 없을 경우 사용한다. 예를 들면, 뉴스 추천 서비스에서 중요 키워드를 추출하여 보관하였을 경우, 단어들의 종류 및 개수를 따로 지정할 수 없기 때문에 집합(Set) 속성 타입을 이용한다.

  · 날짜/시간형(Datetime) : 날짜 및 시간 값을 가지는 경우 사용한다. 예를 들면, TV 추천 서비스에서 프로그램 방영 시작 시간이 이에 속한다.

- 속성에 대한 선호도 계산 방식 : 콘텐츠에 대한 선호도는 콘텐츠가 포함하고 있는 각 속성에 대한 선호도를 계산한 후, 계산된 선호도를 바탕으로 사용하고자 하는 학습 알고리즘을 적용시켜 얻을 수 있다. 속성마다 선호도를 계산하는 방법이 다르며, 계산 방법의 종류는 다음과 같다.

  · 단일값(Unity) : 속성값이 하나일 경우 사용하며, 하나의 선호도 값을 갖는다. 예를 들면, TV 추천 서비스에서 단지 하나의 값만 가지는 채널(channel) 속성이 이에 해당한다.

  · 최대값(Max) : 속성값이 여러 개일 경우에 사용하며, 각 속성값에 대한 선호도 중 최대값을 취한다.

  · 최소값(Min) : 속성값이 여러 개일 경우 사용하며, 각 속성값에 대한 선호도 중 최소값을 취한다.

  · 평균값(Avg) : 속성값이 여러 개일 경우 사용하며, 각 속성값에 대한 선호도 값들의 평균값을 취한다..

  · 코사인 유사도(Cos) : 속성값이 여러 개일 경우 사용하며, 속성 값들 간의 코사인 유사도 값을 취한다.

  · 전체 합산(Add) : 속성값이 여러 개일 경우 사용하며, 각 값에 대한 선호도를 전부 합산하여 속성에 대한 선호도로 나타낸다.

  · 테이블(Table) : 속성값이 여러 개일 경우 사용하며, 선호도를 나타낸 값이 table 형태를 취한다. 예를 들면, TV 추천 서비스에서 시간대별 장르에 대한 선호도, 요일별 채널에 대한 선호도 등이 이에 속한다.

  이 밖에 여러 가지 선호도를 계산하는 방식이 있을 수 있지만, 구현중인 모델링 시스템에서는 위에서 정의한 형식만 사용하였다.

- 속성간의 연관성 : 속성 간에 나타날 수 있는 관계를 표현하기 위한 것이다. 예를 들면, 장르와 서브장르라는 속성을 살펴보면, 'a라는 서브장르는 A라는 장르에 의존한다.'라는 의존성이 존재하며 이를 나타내기 위해 이러한 일반성을 도출하였다. 그러나 어떤 관계성들이 더 존재하는지에 대한 사항은 아직 고려중이며, 현재 구현 중에 있는 사용자 모델링 시스템에서 이 일반성에 관한 내용은 사용하지 않고 있다.

3.1.2 학습 주기를 나타내는 속성의 일반화

사용자 모델을 구성하는 정보를 추출하기 위해서는 학습이 필요하며, 단지 기존의 히스토리를 이용하여 한 번의 학습으로 사용자 모델을 이끌어 내어 끝내는 것이 아니라, 어느 시점까지의 학습된 사용자 모델을 바탕으로, 그 이후의 사용자 이용 히스토리 정보를 통해 업데이트 할 수 있어야 한다. 이런 내용을 반영할 수 있게 일반화하여 정의한 것이 '학습 주기'라는 요소가 되며 학습 주기의 명세를 통해서, 사용자 모델의 갱신 주기를 정의할 수 있는 기준을 정할 수 있다.

3.1.3 히스토리 유지 방식에 관한 속성의 일반화

사용자 이용 히스토리는 사용자 모델을 구축하기 위한 입력 자료로 사용되며, 이러한 히스토리에 대한 유지 기간은 서비스에 따라서 다를 수 있으며, 히스토리 유지에 대한 설정을 일반화할 경우, 다음과 같은 세 가지로 분류해 볼 수 있다.

- 새로운 사용자 모델이 구축 되었을 경우에 삭제되는 경우
- 히스토리 유지기간에 대해 명시적으로 기간을 입력 받을 경우 : 기간에 대한 값을 명세해야 한다.
- 히스토리 정보 전체를 계속 유지시켜야할 경우

3.1.4 콘텐츠 유지 방식에 관한 속성의 일반화

TV에서 하나의 방송 프로그램과 뉴스에서 하나의 기사에 대한 인스턴스는 콘텐츠를 의미한다. 콘텐츠는 선호정보를 타나내는 속성(학습에 필요한 속성)들과 그렇지 않은 속성들을 포함한다. 이런 콘텐츠는 각각의 서비스마다 그 유지기간이 달라질 수 있으며, 유지에 대한 관리가 필요하며, 콘텐츠를 유지하는 기간은 다음과 같이 지정할 수 있다.

- 콘텐츠 유지기간에 대해 명시적으로 기간을 입력 받을 경우 : 이 경우에는 기간에 대한 값을 명세해야 한다.
- 콘텐츠 정보 전체를 계속 유지시켜야할 경우

3.1.5 추천 리스트를 관리하기 위한 속성의 일반화

추천되어질 항목 역시 도메인마다 다르게 적용된다. 예를 들면, TV의 경우에는 시간적으로 추천 서비스를 받는 시점을 기준으로 현재 방송 프로그램과 앞으로 방영될 프로그램에 대한 콘텐츠 항목들로 이루어질 것이고, 뉴스 추천 서비스의 경우 현재 등록된 콘텐츠(기사) 리스트 중 최근 콘텐츠(기사)가 추천되어질 것이다. 이렇게 추천 리스트를 구성하기 위한 항목으로 다음과 같은 설정이 필요할 것이다.

- 기간을 명시적으로 입력 받을 경우 : 이 경우에는 기간에 대한 값을 명세해야 한다.
- 콘텐츠 전체가 대상일 경우 : 기간에 상관없이 콘텐츠 정보 전체를 대상으로 할 경우에 사용된다.

또한 추천될 콘텐츠의 개수를 명세할 수 있는 다음과 같은 속성이 필요하다.

- 추천 되는 아이템의 개수 : 몇 개의 아이템을 추천할지 명세한다.

3.2 사용자 모델링 시스템에 사용된 모듈 및 관련 기술

유비쿼터스 환경에서 개인화 서비스를 지원하기 위한 사용자 모델링 시스템의 대략적인 구조도는 [그림 1]과 같으며, 구조도에서 사용된 선은 데이터의 흐름을 나타낸다.

사용자 모델링 시스템은 모델 스키마 편집 및 생성 모듈, 데이터베이스 연결 모듈, 학습(갱신) 모듈, 추천 엔진, 서버 및 클라이언트 모듈로 구성되며, 각 모듈들의 역할에 대한 자세한 사항은 다음과 같다.

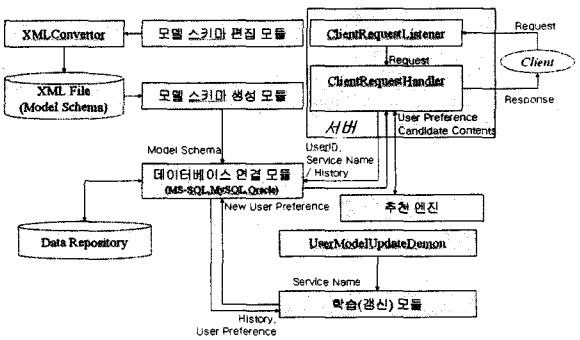

그림 1 사용자 모델링 시스템의 구조도

3.2.1 모델 스키마 편집 모듈

모델 스키마 편집 모듈은 도메인 전문가가 특정 도메인에 대한 개인화 서비스에 필요한 정보를 입력할 수 있게 하며, 이렇게 입력된 정보를 표준 문서인 XML 파일로 생성해 주는 모듈이다. 개인화 서비스에 필요한 정보는 본 논문의 3.1절에서 설명한 일반화된 속성의 실제 값들을 의미하며, 입력된 값에 대한 더 자세한 정보는 4.2절에서 확인할 수 있다. [그림 2]는 구현된 모델 스키마 편집 모듈의 실행화면이다.

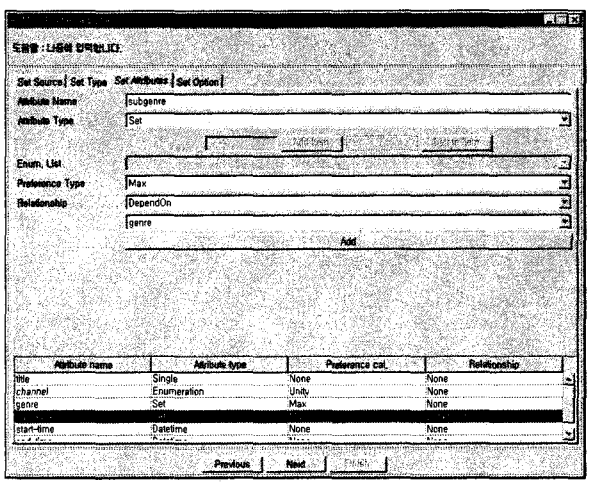

그림 2 모델 스키마 편집 모듈

[그림 3]은 'Ubiquitous Smart Space'라는 도메인에서 DTV 프로그램 추천의 개인화 서비스에 대한 정보를 입력하여 생성한 XML 문서의 일부이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<domain-description
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation=... name="USS">
  <service-description update-period="259200" name="TV">
    <attribute type="Single" name="title">
      <preference-calculus type="None"/>
      <relation type="None"/>
    </attribute>
    <attribute type="Enumeration" name="channel">
      <preference-calculus type="Unity"/>
        <enumeration-list value="KBS2"/>
        <enumeration-list value="MBC"/>
        <enumeration-list value="OCN"/>
        ....
      <relation type="None"/>
    </attribute>
    <attribute type="Set" name="genre">
      <preference-calculus type="Max">
      </preference-calculus>
      <relation type="None"/>
    </attribute>
    <attribute type="Set" name="subgenre">
      <preference-calculus type="Max"/>
      <relation type="DependOn" value="genre"/>
    </attribute>
    <attribute type="Datetime" name="start_time">
      <preference-calculus type="None"/>
      <relation type="None"/>
    </attribute>
    <attribute type="Datetime" name="end_time">
      <preference-calculus type="None"/>
      <relation type="None"/>
    </attribute>
    <attribute type="Enumeration" name="grade">
      <preference-calculus type="None"/>
        <enumeration-list value="0"/>
        <enumeration-list value="7"/>
        <enumeration-list value="12"/>
        ....
      <relation type="None"/>
    </attribute>
    <attribute type="Set" name="keyword">
      <preference-calculus type="Cos"/>
      <relation type="None"/>
    </attribute>
    <keep-contents type="Whole"/>
    <domain-candidate-extraction type="Manual"
     only-not-recommended-item="false" from-period="0"
     to-period="24" number-of-items="5"/>
    <domain-keep-user-history type="Whole"/>
  </service-description>
</domain-description>
```

그림 3 모델 스키마 편집 모듈에 의해 생성된 XML 문서

3.2.2 모델 스키마 생성 모듈

모델 스키마 편집 모듈을 통해 생성된 XML 파일을 입력으로 받는 모델 스키마 생성 모듈은, XML 파서를 통해 입력 받은 XML 파일로부터 개인화 서비스에 대한 정보를 추출한 후, 추출된 정보는 바로 데이터베이스 연결 모듈로 넘겨진다. 넘겨진 정보를 바탕으로 데이터베이스 연결 모듈은 적절한 형태로 레코드를 입력하고, 필요한 테이블을 생성한다.

3.2.3 데이터베이스 연결 모듈

데이터베이스 연결 모듈은 데이터베이스에 관련된 모든 입출력을 담당한다. 따라서 모델 스키마 생성 모듈 및 학습(갱신) 모듈, 추천 엔진 모듈, 서버 모듈이 데이터베이스 연결 모듈과 직접적으로 관련되며, 현재 구현중인 데이터베이스 연결 모듈은 MySQL, MsSQL 및 Oracle 데이터베이스와 연동이 되도록 구현하였다. [그림 4]는 본 논문에서 제안한 사용자 모델링 시스템을 위한 데이터베이스의 논리적 구조를 나타낸 것이다.

3.2.4 학습(갱신) 모듈

모델 스키마 편집 모듈을 통해서 입력되었던 선호 정보에 대한 속성 및 학습 주기에 대한 속성 정보는 데이터베이스에 특정 형식으로 기록되며, 이 정보는 학습(갱신) 모듈에서 사용하여 사용자 모델을 생성하게 된다. 학습(갱신) 모듈에서는, 다양한 학습 알고리즘을 사용자 모델링 시스템에서 쉽게 사용할 수 있도록 하기 위해서 통합된 API를 제공한다. 따라서 개발자는 단지 사용하고자하는 알고리즘을 이 API에 적절하게 변형시켜 사용하면 된다.

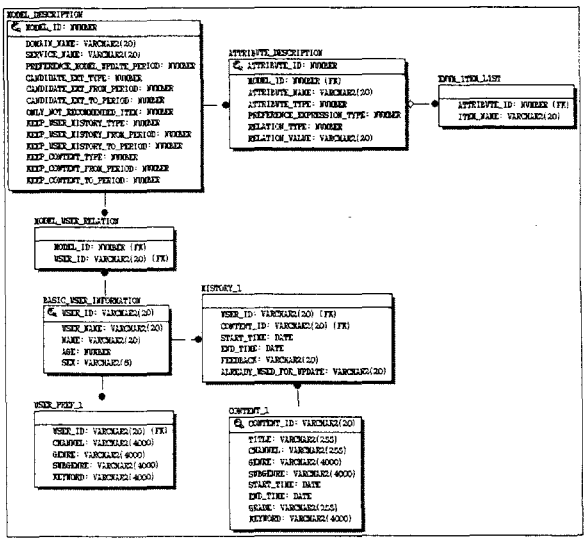

그림 4 사용자 모델링을 통해 생성된 데이터베이스 논리 구조

3.2.5 추천 엔진

추천 엔진은 데이터베이스에 저장된 추천 리스트를 관리하기 위한 속성 정보 및 학습(갱신) 모듈을 통해 생성한 사용자 모델을 사용하며, 추천 알고리즘을 적용해서 특정 도메인에 관련된 콘텐츠의 추천 리스트를 생성한다. 학습(갱신) 모듈과 마찬가지로 추천 엔진은, 다양한 추천 알고리즘을 사용자 모델링 시스템에서 쉽게 사용할 수 있도록 하기 위한 통합된 API를 제공한다.

3.2.6 서버 및 클라이언트 모듈

사용자 모델링 시스템을 구현하는데 있어서, 전체적인 구조는 여러 가지 방법을 사용할 수 있겠지만, 개인화 서비스를 통합적으로 관리하기에 적합한 서버/클라이언트 구조를 채택하였다. 또한 서버와 클라이언트는 XML 문서를 통해서 서로 통신하는 구조를 채택하였다. 서버는 사용자 모델을 생성하기 위한 학습(갱신) 모듈 구동 및 추천 엔진의 구동, 콘텐츠의 입력, 사용자 등록에 관련된 기능을 수행할 수 있게 되어 있고, 다중 클라이언트의 요청을 수행하기 위해 쓰레드 기법을 사용하였으며, 클라이언트는 제공된 클라이언트 측의 API를 통해 서버와 통신하여 서버 측의 기능을 사용할 수 있게 구현하였다.

4 실험 및 결과

4.1 실험 환경

본 논문에서 제안한 사용자 모델링의 실험을 위해 준비한 환경은 [표 1]과 같다.

CPU	펜티엄4 1.8 GHz
O/S	Windows XP Professional
Memory	1 GBytes
HDD	40 GBytes
Language	Java (JDK 1.5)
Database	Oracle 9i

표 1 실험 환경

[그림 2]와 같은 모델 스키마 편집 모듈을 사용하여 DTV 프로그램 추천 서비스에 관한 설계를 하였다. 데이터는 (주) 닐슨 미디어 리서치에서 2006년 1월 1일부터 2006년 1월 31일 동안 수집된 TV 히스토리이며, 학습 및 추천의 입력으로 사용하였다. 시스템에서 제공한 클라이언트 API를 사용하여 클라이언트 모듈을 구현하였고, 클라이언트 모듈에 [그림 5], [그림 6]과 같은 GUI를 사용하여 결과를 쉽게 확인할 수 있게 하였다.

4.2 실험 결과

모델 스키마 편집 모듈을 사용해 입력된 정보는 다음과 같다.

- 도메인명 : USS(Ubiquitous Smart Space)
- 서비스명 : DTV
- 학습(갱신) 주기 : 6 개월
- 입력된 속성에 대한 정보는 [표 2]와 같다.

속성명	속성타입	선호도 계산 타입	관계
title	Single	None	None
channel	Enumeration	Unity	None
genre	Set	Max	None
subgenre	Set	Max	DependOn(genre)
start-time	Datetime	None	None
end-time	Datetime	None	None
grade	Enumeration	None	None
keyword	Set	Cos	None

표 2 USS DTV 추천 서비스를 위한 입력 속성 정보

genre(장르), subgenre(서브장르), grade(등급) 속성의 경우, 모델 스키마 편집 모듈을 통해서 해당하는 열거형 값들은 입력하였다.(예: 장르의 경우, 교양/정보, 취미/레저, 시사/다큐, 드라마, 연예/오락, 교육, 홈쇼핑 등 12개의 값)

선호도 계산 타입이 'None'일 경우 선호도 계산에서는 제외되며, 그 외의 경우 앞서 설명한 선호도 계산 형식을 따른다. 'None'이라는 선호도 계산 타입은 콘텐츠를 구성하는 속성에는 포함되지만, 사용자 모델을 구성하는 선호도 정보로 사용되지 않는 속성을 의미한다.

서브장르 간의 관계는 설정되어 있지만, 구현된 사용자 모델링 시스템에서는 이 관계에 대한 연산은 수행하지 않는다.

- 히스토리 및 콘텐츠는 데이터 전체를 유지하는 방식을 사용하였으며, 추천 리스트는 추천하는 시점으로부터 향후 24시간 내에 있는 콘텐츠 중에 선호도가 높은 상위 5개의 콘텐츠를 추천하게 설정하였다.

위에서 입력한 정보를 바탕으로 클라이언트 모듈에서 특정 사용자에 대한 학습을 수행시킨 결과는 [그림 5]에 나타나 있다. [그림 5]에서 표에 나타난 정보는 첫 칼럼부터 속성명, 해당 장르가 히스토리에 나타난 횟수, 이전 학습된 선호도, 갱신된 선호도, 이전 학습된 선호도와 갱신된 선호도와의 차이를 보여주고 있다.

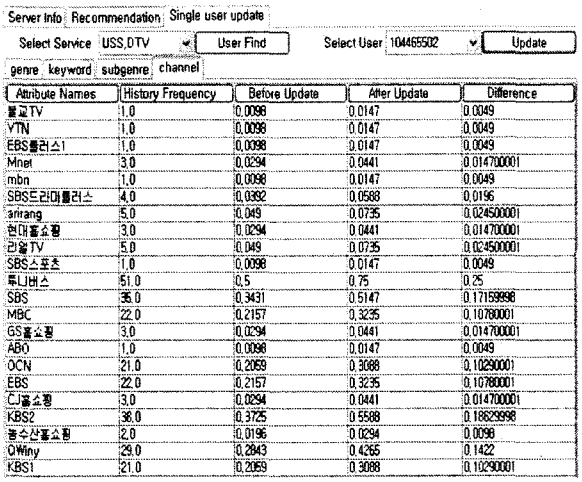

그림 5 학습 결과

[그림 6]는 학습된 사용자 선호도 모델을 바탕으로, 추천엔진을 수행시킨 결과로 추천된 아이템 리스트를 보여주고 있다. 각 아이템은 하나의 콘텐츠, 즉 하나의 DTV 프로그램 인스턴스를 나타낸다. 표에서 마지막 칼럼(Column)에 표현된 것은, 학습으로 계산된 각 콘텐츠에 대한 사용자의 선호도를 나타낸다.

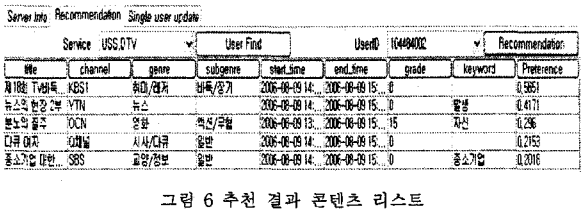

그림 6 추천 결과 콘텐츠 리스트

5. 결론 및 향후 과제

유비쿼터스 컴퓨팅 환경에서는 개인의 특성을 고려한 많은 개인화 서비스들이 동적으로 생성되어 서비스될 것이다. 개인화 서비스를 위해서는 사용자 모델이 일반적으로 사용되지만, 매번 서비스가 구축될 때마다, 사용자 모델을 새로이 설계하고 구현하는 문제가 따른다. 본 논문에서는 이러한 문제를 해결하고 급변하는 유비쿼터스 컴퓨팅 환경에 적용하기 위해서, 개인화 서비스에서 사용되는 정보들의 일반화를 통해서 보다 쉽고 빠른 개인화 서비스의 구축 방법을 제안하였다. 단순히 개인화 시스템에 필요한 정보들을 모델 스키마 편집 모듈을 통해 입력하고, 학습 및 추천을 위해 사용할 알고리즘을 시스템에서 제

공한 API에 적합한 형태로 재구성함으로써 쉽고 빠르게 개인화 서비스가 구축됨을 제시하였다. 또한 구축된 사용자 모델링 시스템은 표준 문서인 XML 파일을 통해 그 정보를 기록 보관하여, 단순히 XML 문서의 교환을 통해 사용자 모델에 관한 정보를 공유할 수 있는 가능성을 제시하였다.

본 논문에서 제안한 사용자 모델링 시스템은 제한적인 개인화 서비스(뉴스 및 TV 추천 개인화 서비스)만을 분석하여, 개인화 시스템의 일반성을 도출하였기 때문에 완전하지 않다. 보다 많은 개인화 시스템의 분석을 통해, 일반화 과정을 반복한다면, 유비쿼터스 컴퓨팅 환경에서 보다 안정적으로 개인화 서비스를 구축할 수 있는 일반화된 방법을 찾을 수 있을 것이다. 그리고 각 개인화 서비스의 사용자 모델에서 사용한 선호 정보를 변환할 수 있는 모듈의 개발과, 현재 자바 애플리케이션으로 구현된 사용자 모델링 시스템을 보다 개방적인 시스템으로 변경하기 위한 웹 서비스로 전환 개발 및 사용자 모델링 연구 분야에서 'Cold start problem'을 해결하기 위해 자주 언급되는 Stereotype에 관한 일반화 작업이 필요하다.

6. 참고문헌

[1]Marc Weiser, "The Computer for the 21st Century" Scientific American, 265(3), 1991

[2]Shlomo Berkovsky, "Ubiquitous User Modeling in Recommender Systems", UM2005, LNAI 3538, 496-498, 2005

[3]Michael Yudelson, Tatiana Gavrilova, Peter Brusilovsky, "Towards User Modeling Meta-Ontology", UM2005, 448-452, 2005

[4]Dominik Heckmann, Antonio Krüger, "A User Modeling Markup Language (UserML) for Ubiquitous Computing", UM2003, 393-397, 2003