



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사 학위 논문

하이브리드 필터링 기반 개인화된 추천
프레임워크의 개발 및 적용

지도교수 조 정 원

제주대학교 교육대학원

컴퓨터교육전공

강 의 영

2007년 2월

하이브리드 필터링 기반 개인화된 추천 프레임워크의 개발 및 적용

지도교수 조 정 원

이 논문을 교육학 석사학위 논문으로 제출함.

2006년 12월 일

제주대학교 교육대학원 컴퓨터교육전공

제출자 강의영

강의영의 교육학 석사학위 논문을 인준함.

2007년 2월 일

심사위원장 김철민 인

심사위원 박찬정 인

심사위원 조정원 인

<국문 초록>

하이브리드 필터링 기반 개인화된 추천 프레임워크의 개발 및 적용

강 의 영

제주대학교 교육대학원 컴퓨터교육전공

지도교수 조 정 원

개인화는 정보화 시대의 정보과잉 속에서 선택의 문제를 해결하는데 도움을 준다. 다시 말하자면 개인화 서비스는 개인의 필요와 목적, 지식, 흥미 또는 상황에 따라 개인별로 다른 정보를 제공함으로써 개인의 시간과 노력의 낭비를 줄이고, 만족감을 얻어낸다. 그러므로 개인화 서비스를 다양한 분야에 적용하면 다음과 같은 긍정적인 효과를 볼 수 있다. 교육에 있어 학습자에게 제공되는 개인화 서비스의 경우에는 학습자 중심의 학습을 제공한다. 이는 학습자의 학습에 대한 결정 동기를 고무시켜 주고, 그에 따라 학업성취에 긍정적인 결과를 가져온다. 또한 기업이 고객을 대상으로 하는 개인화 서비스의 경우에는 고객의 충성도를 높여 재방문과 이용률을 향상시키고 수익을 창출하는 계기가 된다. 그러나 개인화 서비스를 제공하는 입장에서는 방대한 자료를 선별하고, 시스템을 구축하기 위해 많은 시간과 노력을 투자해야 한다. 또한 개인화를 시도하는 대상 아이템의 변화에 맞춰 상응하는 시스템을 구축하려면 비슷한 투자가 반복되어야 한다.

따라서 본 논문에서는 대상 아이템의 변화에 관계없이 재사용이 가능한 개인화된 서비스를 지원하는 ‘하이브리드 필터링 기반 개인화된 추천 프레임워크’를 제안한다. 제안된 프레임워크는 각 아이템별로 구조화된 속성들을 가지고 사용자의 선호정보에 따라 개인화된 정보를 제공한다. 이를 위해 본 연구는 기존의 개인화 관련 시스템들을 분석하고, 요소 기술들을 이용하여 프레임워크를 개발한다. 그리고 제안된 프레임워크의 적용으로 개인화된 수강과목 추천 시스템과 개인화된 관광지 추천 시스템을 설계하고 구현하였다. 그리고 적용 시스템과 프레임워크의 성능을 평가하고 분석하여 본 프레임워크의 우수성을 보인다.

※ 본 논문은 2006년 12월 제주대학교 교육대학원 위원회에 제출된 교육학 석사학위 논문임.

목 차

제 1 장. 서 론.....	1
1.1 연구의 배경 및 필요성.....	1
1.2 연구 주제 및 연구 방법.....	2
제 2 장. 관련 연구 및 동향.....	3
2.1 개인화의 의미.....	3
2.2 개인화된 추천을 위한 정보 필터링.....	4
2.2.1 내용기반 필터링.....	4
2.2.2 협동적 필터링.....	4
2.2.3 규칙기반 필터링.....	5
2.2.4 기타 필터링.....	5
2.2.5 혼합형 필터링.....	6
2.3 기존의 프레임워크와 추천 시스템.....	6
제 3 장. 하이브리드 필터링 기반의 개인화된 추천 프레임워크.....	9
3.1 프레임워크의 개요.....	9
3.2 프레임워크의 구조.....	10
3.2.1 퍼시스턴스(Persistence) 계층.....	11
3.2.2 비즈니스(Business) 계층.....	12
3.2.3 사용자 인터페이스(User Interface) 계층.....	19
제 4 장. 프레임워크의 적용.....	20
4.1 시스템 구현 환경.....	20

4.2 개인화된 수강과목 추천 시스템.....	22
4.2.1 데이터 속성 구조.....	22
4.2.2 시스템의 설계.....	23
4.2.3 시스템의 구현.....	27
4.3 개인화된 관광지 추천 시스템.....	28
4.3.1 데이터 속성 구조.....	28
4.3.2 시스템의 설계.....	29
4.3.3 시스템의 구현.....	30
제 5 장. 성능 평가.....	33
5.1 수강과목 추천 시스템 성능 평가.....	33
5.2 관광지 추천 시스템 성능 평가.....	37
5.3 프레임워크 성능 분석.....	38
제 6 장. 결 론.....	41
참고 문헌.....	42
<Abstract>.....	45

표 차례

<표 1> 시스템 구현 환경.....	20
<표 2> 수강과목 추천 시스템의 적중률.....	34
<표 3> 수강과목 추천 시스템의 평균 재현율.....	36
<표 4> 수강과목 추천 시스템의 평균 정확도.....	36
<표 5> 관광지 추천 시스템의 적중률.....	37
<표 6> 관광지 추천 시스템의 평균 재현율.....	38
<표 7> 관광지 추천 시스템의 평균 정확도.....	38
<표 8> 프로그램 코드 개발 생산성 비교.....	40

그림 차례

<그림 1> CXMS의 주요 계층 구조.....	7
<그림 2> 하이브리드 필터링 기반 개인화된 추천 프레임워크의 계층 구성 및 역할.....	11
<그림 3> 비즈니스 계층 구조.....	12
<그림 4> <i>Recommender</i> 자식 모듈 사용의 일반적인 구성.....	13
<그림 5> 유사도 계산 관련 모듈 구성.....	14
<그림 6> 유사 사용자들의 내역(History) 검사 모듈 구성.....	15
<그림 7> 규칙 생성과 규칙 검사 모듈 구성.....	16
<그림 8> <i>Rule</i> 클래스의 필드 정보.....	16
<그림 9> <i>RuleUnit</i> 클래스의 필드 정보.....	17

<그림 10> <i>ConObject</i> 와 <i>ConObjectUnit</i> 의 내부 구조.....	18
<그림 11> 조건부 검사 방법.....	18
<그림 12> 스트럿츠(Struts)의 전체 구조.....	21
<그림 13> 재구성한 학생 속성.....	23
<그림 14> <i>Student</i> 와 <i>Subject</i> 클래스의 필드(Fields) 구조.....	24
<그림 15> 개인화된 수강과목 추천 시스템 구조도.....	25
<그림 16> 수강과목에 대한 내용기반 필터링의 적용.....	26
<그림 17> 수강과목 추천을 위한 입력과 출력 화면.....	27
<그림 18> 재구성한 사용자 속성.....	28
<그림 19> 개인화된 관광지 추천 시스템 구조.....	29
<그림 20> <i>Poi</i> 와 <i>User</i> 클래스의 필드(Fields) 구조.....	29
<그림 21> 회원 등록 중 관광지 선호 정보 입력.....	31
<그림 22> 관광지 추천을 위한 입력과 출력 화면.....	32
<그림 23> 수강과목 추천 시스템의 성능 평가.....	34
<그림 24> 관광지 추천 시스템의 성능 평가.....	37

제 1 장. 서 론

1.1 연구의 배경 및 필요성

최근 교수-학습의 패러다임을 살펴보면, 일방적인 교사와 교육 행정 편 중심에서 점차 학생 선택의 폭이 넓어지는 학습자 중심의 교육 방향으로 나아가고 있다[1]. 그리고 이러한 흐름을 뒷받침해 주기 위해 다양한 교육과정이 생겨났고, 이와 더불어 정보 기기의 발달은 방대한 양의 교육 자료들을 양산했다. 이로 인해 교육과정과 교육 자료들에 대한 선택의 주체인 학습자들은 선택의 딜레마에 놓이게 된다. 이에 학습자의 선택에 도움을 줄 수 있는 개인화 기술의 도입이 필요하다. 최근 이런 분위기에 맞추어 교육 분야에 개인화를 적용한 연구 사례가 많이 있다. 개인화 기술은 교육 과정, 교육 자료[2] 및 교수-학습 방법[1][3] 등과 같이 개인화가 필요한 교육 분야에 다양한 아이템을 가지고 적용이 가능하다. 또한 개인화 기술은 교육 분야뿐만 아니라 일반 산업 분야, 즉 기업의 입장에서 고객 충성도와 재방문율을 높여 수익을 창출하는 계기를 만드는 등의 긍정적인 효과가 있다. 그러나 이러한 개인화 서비스를 제공하는 입장에서는 방대한 자료를 선별하고, 추천 전략에 맞게 시스템을 구축하기 위해 많은 시간과 노력을 투자해야 한다. 또한 개인화를 시도하는 대상 아이템의 변화에 맞춰 상응하는 시스템을 구축하려면, 아이템들의 데이터 속성 구조를 잡고, 그에 종속된 시스템 구조를 처음부터 새롭게 구축해야 하기 때문에 비효율적으로 중복된 작업이 나타난다. 하지만 데이터 속성 구조를 구축하는 부분과 추천 전략에 따라 추천을 처리하는 부분을 분리하는 방식을 통해 추천시스템을 구축하면 각 분야의 전문성을 살려 집중적으로 데이터 속성을 연구하면서, 필요한 추천 전략에 따라 효율적으로 시스템을 구성하는 것이 가능하다. 따라서 본 연구에서는 교육 분야에 한정적으로 적용하기 위한 개인화된 추천이 아니라 범용(凡庸)적인 목적으로 개인화된 추천 프레임워크, 그 중에서도 하이브리드 필터링 기반의 개인화된 추천 프레임워크(이하 프레임워크라 함.)를 개발하고 이를 적용한 사례를 보임으로써 향후 교육 분야, 그리

고 좀 더 일반화하여 개인화가 필요한 여러 분야에서 개인화된 추천 시스템을 구축하는데 도움을 주고자 한다.

1.2 연구 주제 및 연구 방법

본 연구의 주제는 개인화된 추천 아이템의 특성을 고려하여, 재사용가능한 프레임워크를 개발하고, 이를 기반으로 추천시스템에 적용해봄으로써 보다 손쉽게 개인화된 추천 시스템을 구축할 수 있는 기반을 마련하는데 있다. 제시한 본 연구의 주제에 따라 본 연구는 다음과 같이 수행되었다.

- 첫째, 개인화된 선행 연구들의 특징과 문제점 그리고 개선 방안에 대해 고찰한다.
- 둘째, 프레임워크의 기능과 구조를 명확히 구체화하여 제안한다.
- 셋째, 프레임워크를 활용하여 수강 과목 추천 시스템과 관광지 추천 시스템을 구축한다.
- 넷째, 각 추천 시스템의 적중률을 측정하여 시스템의 성능을 분석하고, 평가하여 프레임워크의 적용 성능을 분석한다.

이를 위해 본 논문은 아래와 같이 구성한다. 제 2장에서는 기존의 문헌연구와 응용서비스 연구를 통해 개인화의 개념과 특징 그리고 문제점을 고찰한다. 제 3장은 본 연구에서 제안하는 프레임워크의 구조와 기능을 제시한다. 제 4장에서는 프레임워크를 기반으로 수강 과목 추천 시스템과 관광지 추천 시스템을 설계하고 구현한다. 제 5장에서는 4장에서 개발한 프레임워크와 구현한 추천 시스템들의 성능을 평가한다. 제 6장에서는 본 연구를 정리하고, 향후 연구 방향에 대하여 논한다.

제 2 장. 관련 연구 및 동향

2.1 개인화의 의미

개인화는 사용자에게 개인의 필요와 목적, 지식, 흥미 또는 상황에 맞는 상품과 서비스 그리고 정보를 제공한다[4][5]. 개인화 서비스의 과정을 통하여 개인 사용자는 자동적으로 선별된 정보만을 제공받음으로써 정보화 사회의 정보과잉 문제를 완화하고, 시간의 절약을 가져온다. 개인화 서비스를 제공하려면 크게 두 가지를 고려해야한다. 하나는 프로파일을 수집하는 방법이고, 또 하나는 수집된 정보를 가지고 ‘어떻게’ 개인 사용자가 원하는 정보를 전달할 것인가 하는 부분이다. 이를 설명하기 위해 개인화 서비스의 흐름을 표현하면 다음과 같다. 첫째, 사용자와 데이터 정보의 수집이 이루어진다. 이 때 수집이 이루어지는 방식은 크게 3가지이다.

- 명시적 방법(Explicit profiling) : 사용자에게 직접 정보를 요구한다.
- 묵시적 방법(Implicit profiling) : 로그 파일이나 구매 내역 등을 기록하여 사용자가 의식하지 못한 상태에서 정보를 수집한다.
- 남겨진 데이터(Using legacy data) : 사용자가 어떤 행위나 입력을 행한 뒤 부수적으로 얻게 되는 데이터를 의미한다. 예를 들어 신용카드를 자주 사용하여 결제가 완료된 사용자는 구매력이 높은 사용자라는 정보를 얻는다[6].

둘째, 수집된 정보를 가공하여 개인화 서비스 모듈에 전달할 준비를 한다. 이는 위에서 언급한 ‘어떻게’의 과정의 입력으로 들어가기 위한 최종 준비 단계이다. 셋째, 개인화 서비스 모듈이 동작된다. 이 때 앞 단계에서 가공된 수집 정보가 입력 데이터로 활용된다. 넷째, 개인화된 정보가 나오면 개인 사용자에게 전달된다. 다섯째, 개인 사용자는 제공받은 정보에 대한 응답(혹은 평가)을 시스템에 보낸다. 이로써 일련의 과정이 마무리된다. 따라서 어떤 프로파일을 입력받아 어떻게 처리할 것인가가 개인화 서비스를 위한 시스템의 중요한 부분이다.

2.2 개인화된 추천을 위한 정보 필터링

개인화 서비스는 사용자에게 적합한 정보의 추천으로 이루어진다. 개인 사용자에게 특정 정보를 추천하기 위해 가변적이고 방대한 데이터를 정제하는 과정이 필요하다. 이를 정보 필터링이라 한다[7].

정보를 여과하는 필터링 기법들은 다양하다. 추천 접근 방식에 따라 내용기반 필터링(Content-based Filtering), 협동적 필터링(Collaborative Filtering), 규칙기반 필터링(Rule-Based Filtering), 인구통계학적 필터링, 사례기반 추론 등으로 나뉘 볼 수 있다. 그리고 각 필터링 기법의 특징은 다음과 같다.

2.2.1. 내용기반 필터링(Content-based Filtering)

사용자의 프로파일과 추천 아이템의 특징 사이의 비교를 기반으로 항목을 추천한다. 즉, 아이템 항목과 사용자의 선호도 사이의 유사도를 기반으로 하여 추천이 이루어진다. 이는 사용자의 프로파일을 통해 과거의 구매내역이나 추천 결과를 쉽게 반영한다는 장점이 있다. 그러나 서로 다른 아이템이라 하더라도 특징을 나타내는 내용집합(set)이 동일한 상태일 때 두 아이템의 구분(well-written article or badly written one)을 못한다는 단점이 있다. 또한 처음 시스템을 사용하는 사용자에게 대한 초기 사용자 정보가 없다면 추천이 불가능한 초기 사용자 문제가 발생한다[8].

2.2.2. 협동적 필터링(Collaborative Filtering)

협동적 필터링은 사용자의 관심 사항을 예측하기 위해 유사 사용자의 의견을 사용하는 방법이다. 이 방법은 익명(Anonymous) 개인화의 추세에 맞게 개인정보 보호가 용이하다. 사용자의 개인정보 공개 없이 사용자의 취향이 유사한가를 기준으로 보기 때문이다. 또한 숨겨진 선호 패턴을 발견하기가 용이하다. 그러므로 사용자가 좋아하는 논리적 근거가 명백하지 않은 영역에 적용하기 알맞다. 예를 들어, 음악CD나 서적

[9], 영화[10]과 같은 영역에서 많이 응용된다[11]. 하지만 이 필터링은 사용자와 항목 수에 비례해서 계산 시간이 증가한다는 한계가 있다. 그리고 새로운 항목이 등장하여 예측에 기반이 되는 사용자들의 평가가 없을 경우, 해당 항목에 대한 예측을 제공할 수 없는 초기 평가 문제가 존재한다. 그리고 많은 정보 도메인에서는 항목들의 개수는 개별 사용자들이 소화할 수 있는 개수를 훨씬 초과한다는 희소성 문제를 지니고 있다.

2.2.3. 규칙기반 필터링(Rule-based Filtering)

대부분의 최근 연구문헌들에서는 필터링을 위 두 가지의 필터링(내용기반과 협동적 필터링)으로 나누고 있다[12][13]. 그러나 일부에서는 규칙기반 필터링을 별도로 구분하기도 한다. 이 필터링은 사용자 행동 패턴을 추출하여 일정한 규칙을 만들고 해당 규칙에 따라 아이템을 추천하는 방법이다. 의사결정 트리의 작성 시간이 오래 걸리지만 정확한 규칙만 제공된다면 추천의 정확도도 높아진다. 따라서 상품 수가 적고, 고려할 요소가 많은 상품에 응용된다. 금융상품이나 여행상품과 관련된 아이템이 이에 속한다. 그러나 세세한 부분까지 조건을 명시하여 필터링 해야 한다는 한계를 가진다. 따라서 실생활 응용 상황에 유연하게 적용하기 어렵다. 그러므로 규칙을 기반으로 추론과 학습을 하는 지능형 에이전트를 개발하는 방향으로 발전하는 경향을 보인다[14][15][16].

2.2.4. 기타 필터링

앞서 살펴본 필터링 외에도 나이, 성별, 지역과 같이 특정 항목을 선호하는 사용자의 유형을 구분하여 해당 소속에 맞게 필터링을 사용하는 인구통계학적 필터링과 과거에 이미 해결한 문제들을 기반으로 현재의 문제를 해결하는 사례기반 필터링 [17][18]이 있다. 이 필터링은 유사집단의 평가정보를 이용하지 않고, 개인별 속성에 대한 가중치와 속성 값을 이용하여 정보 추천이 가능하고, 사용자가 선호하는 아이템

유형에 대해 피드백 받은 속성 값을 추천에 반영한다는 장점이 있다. 하지만 사례 구축이 어렵고, 일정 시간이상 관찰한 후에야 정보 제공이 가능하다는 한계를 가지고 있다. 또한 필터링은 구현 기술에 따라 수식 기반의 휴리스틱기반과 학습하는 모델기반의 필터링 기술로 나누기도 한다[8].

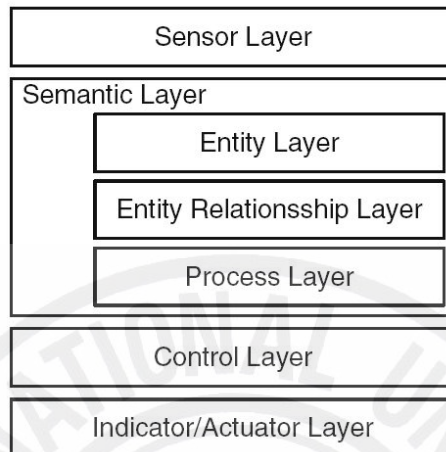
2.2.5. 혼합형 필터링(Hybrid Filtering)

앞서 언급한 바와 같이 각 필터링들은 장, 단점을 가지고 있다. 협동적 필터링의 경우 데이터 블러링 기법으로 속성 정보의 확률분포를 이용하여 알려지지 않은 값을 예측함으로써 희소성의 단점을 극복하려는 연구가 있다[19]. 그러나 추가되는 속성 정보의 양이 너무 작을 경우 정확도를 보장하지 못한다. 따라서 최근에는 단일 필터링을 사용하기보다는 두 가지 이상의 필터링 기법을 혼합하여 사용하는 하이브리드 방식을 취한다[7][20]. 하이브리드 방식은 3가지 형태가 존재한다. 첫째, 서로 다른 필터링으로 서로 다른 결과물을 보여주는 방식이 있다. 둘째, 한 필터링의 과정을 거친 후 다음 필터링으로 넘어가는 방법이다. 셋째, 완전히 혼합된 방식으로 두 필터링이 완전히 혼합되어 기술적인 구분 없이 처리하여 하나의 결과물을 내놓는 것이다 [8]. 본 연구에서 제시하는 하이브리드 방식은 두 번째와 유사하다. 따라서 단일 필터링의 한계를 극복하고, 결과물을 단일화하여 선택의 혼동을 줄였으며, 필터링 기법을 단계적으로 분리하여 적용하였으므로 향후 변형이나 확장이 용이하다.

2.3 기존의 개인화 프레임워크와 추천 시스템

프레임워크란, 특정한 애플리케이션을 제작하는 데 이용할 수 있는 재사용 가능하고(reusable), 반완전한(semi-complete) 형태의 애플리케이션이다[21]. 프레임워크들 중 개인화와 관련된 기존 연구들은 다음과 같다. [4]는 상황인식(Context-aware)을 이용하여 문서들의 내용을 분석하고 관리하여 개인에게 필요한 정보를 제공해 주는

관리도구인 CXMS(Context Management Framework)를 제안한다. 이 시스템은 (그림1)과 같이 4개의 계층을 두고 개인화된 서비스를 제공한다.



(그림 1) CXMS의 주요 계층 구조

자료: A. Zimmermann, M. Specht, and A. Lorenz(2005), Personalization and Context Management, User Modeling and User-Adapted Interaction, Vol. 15, No 3-4, p.283.

센서 계층(Sensor Layer)은 정보 수집기(Information collector)의 역할을 수행하는 부분이다. 이 계층에서는 사용자의 행동이나 상황 변화 등을 감지한다. 시맨틱 계층(Semantic Layer)은 수집된 정보들을 이용하여 상황 모델(context model)을 정의한다. 이 때 상황 모델은 현재의 상황(situation), 즉 사용자의 선호도, 관심, 사회적 지위와 물리적, 기술적 환경을 가지고 작성된다. 제어 계층(Control Layer)은 모델이 특정 상황을 만족했을 때 어떤 행동을 할 것인가를 제어하게 된다. 마지막으로 지시자/행동자 계층(Indicator/Actuator Layer)은 개인화된 정보를 직접 연결하여 사용자에게 특정 문서를 제공하는 계층이다. CXMS는 각 계층별로 의미 있는 구성과 역할을 가지고 있지만, 개인의 행동과 상황에 따른 규칙적인 흐름에 기반하고 있고, 디지털화된 문서의 도메인을 대상 아이템으로 삼고 있다. 따라서 아이템의 변화에 적절히 대응하기에는 부적절하다는 한계를 갖고 있다.

[22]는 3가지 필터링을 위한 프레임워크를 제안하고 있다. 이를 이용하여 식당을

추천하는 예를 들고 있지만, 이 연구는 필터링 기술에 기반을 둔 전체적인 추천 시스템을 만들기 위한 프레임워크라기 보다는 각각의 필터링 기술과 조합을 통한 추천의 예를 보여주고 있다. 따라서 하나의 시스템을 위한 프레임워크라고 보기에는 시스템의 완성도를 높여주기 위한 구조적인 틀이 빈약하다.

기존의 추천 시스템은 앞서 2.2에서 살펴본 기술들과 아이템들의 조합을 위한 테스트 형식으로 구현된다[23][24][25]. 2.2.5절의 앞부분에서 언급된 [23]은 해당 기술을 이용하여 영화 추천 시스템을 구현한다. 하지만, 앞서 지적한 바와 같이 데이터 평가 집합에 따라 초기 사용자 문제와 희소성 문제를 여전히 가지고 있고, 사용자가 선호하는 특정 내용에 기반을 둔 추천은 이루어지지 못한다는 문제를 가지고 있다. [24]는 개인의 정태적 정보와 동태적 정보를 입력받아 추론 규칙을 통해 분류된 타입별로 여행코스를 추천하는 프로토타입의 시스템을 구현한다. 그러나 이 시스템은 개인사용자의 개별 입력 후에 군집화 시킴으로써 집단별로 단일한 결과를 출력하게 되므로, 온전한 개인화 서비스라고 보기에는 한계가 있다. 또한 관광코스에 대해 분석된 규칙에 종속된 시스템이므로 타 아이템으로의 확장에 한계를 가지고 있다. [25]는 고객의 상품 검색과 구매 내역을 통한 고객의 선호도를 파악한 후에 지능적 에이전트를 통해 상품을 추천하는 시스템이다. 사용자에게 직접 정보를 요구하는 것이 아니라 사용자의 로그를 이용하여 선호 정보를 축적하는 묵시적 방법을 통해 정보를 수집한다는 점이 장점이다. 하지만 초기 사용자는 자신의 선호도에 맞는 상품추천이 불가능하다는 한계를 가지고 있고, 구매 환경이 아닌 다른 추천 아이템을 위해서 시스템 전반에 걸쳐 새로운 설계 방안이 필요하다.

제 3 장. 하이브리드 필터링 기반 개인화된 추천 프레임워크

3.1 프레임워크의 개요

기존의 연구들이 제안한 추천 시스템은 단일 필터링에 치우쳐 있거나 영화 추천과 같이 하나의 주제를 가진 응용서비스에 종속되어 있다. 단일 필터링은 혼합형 필터링 방식보다 많은 단점을 드러내고 있고, 응용서비스에 종속된 시스템은 재사용성을 떨어뜨린다. 새로운 아이템에 적응하기 위해서는 처음부터 다시 아이템의 메타데이터에 종속적인 여러 모듈들을 필요로 한다. 이는 개발 노력이나 시간에 따른 비용의 낭비를 초래한다. 따라서 본 논문에서는 단일 필터링의 단점을 보완하고, 추천 대상 아이템의 특성에 적응적이면서 독립적인 모듈들로 개인화된 추천 시스템을 구성하기 위해 하이브리드 필터링 기반 개인화된 추천 프레임워크를 제안하고자 한다(이하 프레임워크라 함). 본 연구에서는 프레임워크를 이용하여 매번 중복되는 사항들의 낭비를 줄인다. 이에 다음과 같은 요구사항을 기준으로 프레임워크의 목적을 구체화한다.

첫째, 데이터 속성 구조와 데이터 추천을 처리하는 모듈을 분리해야 한다. 2.1절에서 언급한 바와 같이 일반적으로 개인화된 추천 시스템을 구축하기 위해 ‘추천 아이템’을 ‘어떻게’ 추천해야 하는가를 중요하게 고려한다. 추천 아이템과 처리 방식이 강력한 연결을 가지고 있으면 시스템을 처음부터 매번 새롭게 구축해야 한다.

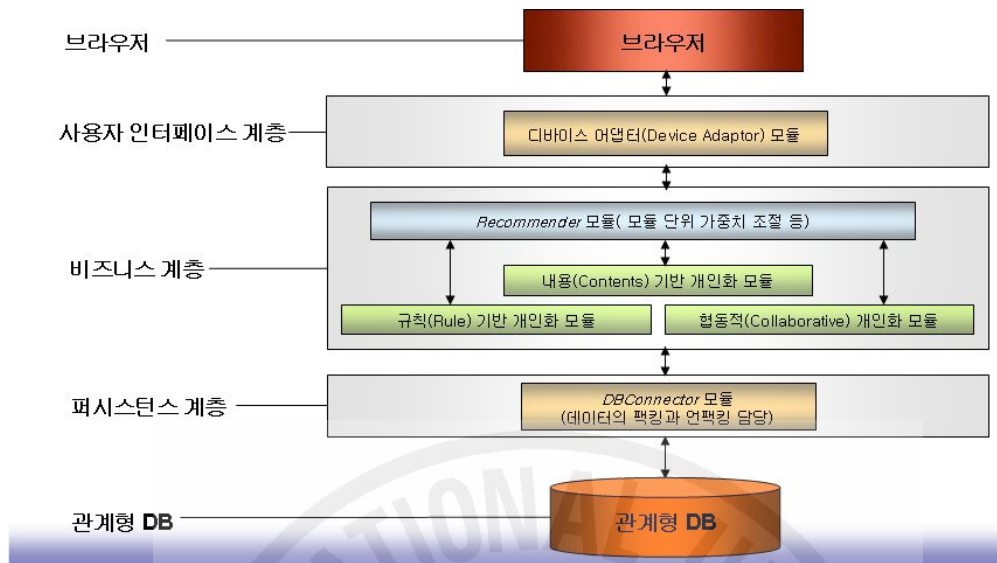
둘째, 추천을 처리하는 기술이 확장된 모듈을 수용해야 한다. 기존 연구에서 살펴본 것과 같이 개인화를 실현하기 위한 기술은 다양하다. 필터링을 이용한 기법도 수많은 수식과 방법들이 존재한다. 이 모든 필터링 기술들을 단일화된 모듈로 구성하는 것은 힘들다. 따라서 다양한 필터링 기술이 수용되고 확장될 수 있는 방안이 필요하다.

셋째, 추천 대상에 따라 추천을 처리하는 과정이 다양하게 제어되도록 고려해야

한다. 아이템의 특성에 따라 요구되는 필터링 기술과 적용 방법은 달라진다. 좋아하는 논리적 근거가 명백하지 않은 분야는 협동적 필터링 과정이 필요하고, 금융 투자와 같이 세부고려사항이 많은 분야는 규칙 필터링 과정이 필요하다는 것과 맥락을 같이 한다. 이에 본 연구에서는 이러한 점들을 고려하여 다음과 같이 프레임워크를 설계한다.

3.2 프레임워크의 구조

본 논문에서 제안하는 프레임워크는 (그림2)와 같이 크게 3가지 계층으로 구성되어 있다. 퍼시스턴스(Persistence) 계층은 데이터베이스에 저장된 데이터들을 본 프레임워크에서 자유롭게 이용할 수 있도록 팩킹(Packing)과 언팩킹(Unpacking)의 역할을 담당하고 있다. 비즈니스(Business) 계층은 본 프레임워크의 핵심 계층으로 하위 계층으로부터 제공되는 데이터들을 연산하여 의미 있는 정보, 즉 추천 데이터를 생성하는 기능을 한다. 본 계층의 중심은 *Recommender* 모듈이다. 이 모듈이 비즈니스(Business) 계층을 관리하며, 이하 모듈 단위의 가중치와 우선순위를 조절한다. 마지막으로 개인 사용자에게 정보를 전달하는 사용자 인터페이스(User Interface) 계층이 있다. 이 계층을 통해 개인 사용자의 단말에 따라 적응적인 사용자 인터페이스로의 확장이 가능하다.



(그림 2) 하이브리드 필터링 기반 개인화된 추천 프레임워크의 계층 구성 및 역할

3.2.1. 퍼시스턴스(Persistence) 계층

팩킹(Packing)을 통해 데이터베이스에 저장되어 있던 데이터를 가공하여 아이템에 적응적인 데이터 속성의 구조를 유지하여 비즈니스 계층 내의 연산 과정을 용이하게 해준다. 또한 언팩킹(Unpacking) 과정을 통하여 비즈니스 계층에서 생산되거나 전달된 여러 데이터들을 데이터베이스에 저장 가능한 형태로 만들어 주는 역할을 한다.

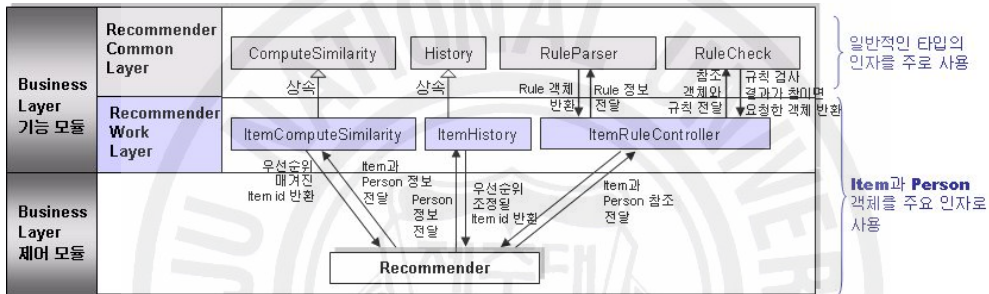
본 계층에서 하는 역할을 살펴보면 다음과 같다.

- **데이터베이스 연결 설정 제어** : 특정 데이터베이스로의 연결을 위한 사용자 정의 및 포트 정보를 가지고 데이터베이스 드라이버를 제어한다.
- **저장 아이템의 데이터 관리** : 저장된 아이템의 정보를 응용 서비스에서 정의한 데이터 속성 구조에 맞도록 재구성하여 생성하고, 요청에 따라 수정 또는 삭제하며, 특정 데이터를 검색하여 프레임워크에 데이터를 가공하여 넘겨주는 기능을 담당한다.

- **개인사용자의 프로파일 데이터 관리** : 개인사용자의 프로파일을 가지고 응용 서비스의 요청에 따라 생성, 수정, 삭제 및 검색을 통하여 데이터베이스에 저장된 데이터를 가공하여 프레임워크에 넘겨주는 기능을 맡는다.

이와 같이 프레임워크와 데이터베이스의 통신을 위해서는 이 계층을 반드시 거쳐야만 접근할 수 있다. 이는 데이터베이스로부터의 독립성을 유지시켜 주는 기능을 해준다.

3.2.2. 비즈니스(Business) 계층

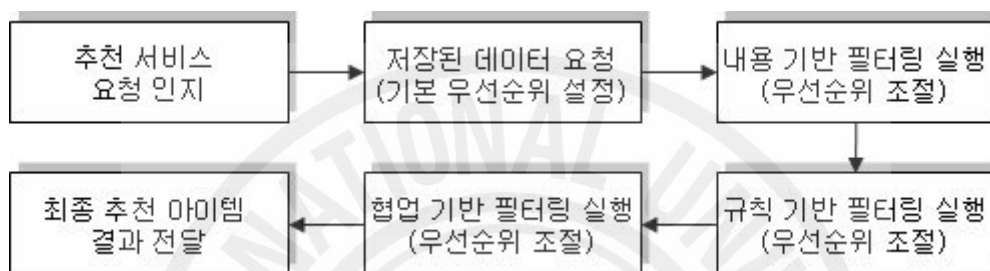


(그림 3) 비즈니스 계층 구조

비즈니스(Business) 계층은 프레임워크의 실제적인 추천 연산을 맡는 핵심적인 계층이다. 본 계층에서는 내용기반 필터링, 협동적 필터링, 규칙기반 필터링과 같이 3종류의 필터링을 이용하여 추천이 이루어진다. 이에 *Recommender*는 이 필터링들을 제어하는 역할을 담당한다. 또한 본 계층은 다양한 아이템들의 데이터 속성 구조에 적응적으로 대처하기 위해 (그림3)과 같이 2개의 세부계층으로 나누어진다. 공통(common) 계층은 여러 아이템들을 필터링 하고자 할 때 공통적으로 사용되는 상위 모듈들로 구성된다. 그리고 작업(work) 계층은 상위 계층에서 상속받거나 객체를 생성하여 활용하는 모듈들로 구성된다. 따라서 추천하는 대상 아이템들에 따라 *Item* 모듈들은 데이터 특성에 따라 *Recommender*의 자식 모듈에 의해 다양하게 적용된다. 비즈니스(Business) 계층의 모듈 구성과 역할을 다음과 같다.

- **Recommender** 모듈

프레임워크의 핵심적인 부분으로써 개인 사용자의 요청을 받아드리고, 퍼시스턴스(Persistence) 계층에 데이터를 요청하며, 각 필터링들을 통해 얻어진 추천 결과들의 가중치 및 우선순위를 부여하여 최종 추천 결과를 도출하는 역할을 맡아 간단하면서도 추천 시스템의 중추적인 역할을 수행하는 메소드들(Methods)로 구성한다. 그리고 이를 상속한 자식 모듈을 통해 추천 전략에 따라 (그림4)의 예와 같이 사용 가능하다.



(그림 4) *Recommender* 자식 모듈 사용의 일반적인 구성

- **내용기반 필터링 관련 모듈**

내용기반 필터링을 하는 방법으로 아이템에 관한 키워드로 필터링을 하는 방법이 있지만 아이템이 완전히 해당 키워드와 일치하거나 연결 관계가 명시되지 않는 경우는 추천이 불가능하다는 한계를 가지고 있다. 따라서 키워드 방식이 아닌 유사도를 이용하여 사용자가 원하는 정보를 필터링 하고자 한다. 개인 사용자의 선호도 정보와 해당 아이템의 특성 정보를 이용하여 개인 사용자와 아이템 사이의 유사도를 계산한다. 이 때 본 프레임워크에서 제공하는 기본적인 유사도 계산식은 식 (1)과 같이 코사인 유사도[26]을 응용한 계산식이다.

$$VS(\vec{P}_u, \vec{A}_i) = \cos\theta = \frac{\vec{P}_u \cdot \vec{A}_i}{|\vec{P}_u| \times |\vec{A}_i|} \quad (1)$$

$$VS(\vec{P}_u, \vec{A}_i) \times \frac{|\vec{P}_u|}{|\vec{A}_i|} = \frac{\vec{P}_u \cdot \vec{A}_i}{|\vec{A}_i| \times |\vec{A}_i|} \quad (\text{단}, |\vec{P}_u| \leq |\vec{A}_i|) \quad (2)$$

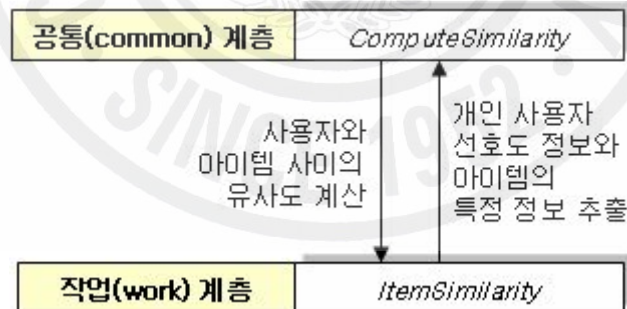
\vec{P}_u : 사용자 u 의 선호 벡터

\vec{A}_i : 아이템 i 의 속성 벡터, 해당 속성 단위의 최고 값으로 고정

$\cos\theta$: 사용자 u 와 아이템 i 사이의 패턴 유사도

식 (1)에서 벡터 유사도는 두 벡터들 사이의 코사인 값을 가지고 유사도를 측정한다. 이 경우에 각 요소 벡터들 각각의 유사도가 아닌 하나의 요소 벡터 집합과 다른 요소 벡터 집합 사이의 패턴의 유사성만 계산된다. 이 문제를 해결하기 위해 유사도 계산식에서 벡터들의 길이를 이용하여 식 (2)와 같이 변형된 계산식을 이용하였다. 유사도 계산 결과는 1에 가까울수록 사용자 선호도와 아이템의 특성정보가 유사하다는 것을 의미한다.

그러나 필요에 따라 *ItemSimilarity*에서 원하는 유사도 계산식을 재정의 하여 사용하는 것도 가능하다. 내용 기반 필터링을 구성하는 두 개의 주요 모듈을 나타내면 (그림5)와 같다.



(그림 5) 유사도 계산 관련 모듈

- 협동적 필터링 관련 모듈

군집화를 통하지 않고 개인 사용자와 다른 개별 사용자 사이의 내역을 이용하여

개인사용자에게 (그림6)와 같이 아이템을 추천한다. 이를 위해 개인 사용자와 다른 개별 사용자 내역 간의 동일 집합 정도와 순서성을 계산한다. 동일 집합 정도는 개인 사용자와 타사용자 사이에 일치하는 내역의 집합의 정도를 의미한다. 그리고 순서성은 개인 사용자와 타사용자의 내역들 중 시퀀스까지 일치하는 정도를 의미한다. 이용 내역이 임계치 이상의 동일한 집합과 동일한 순서를 유지한 타사용자를 유사사용자로 판단하여 그 사용자의 내역을 참고로 추천을 한다. 그리고 이 임계치는 아이템 특성과 사용자 집합에 따라 *Recommender*의 확장모듈을 통해 변경이 가능하게 하여 추천 시스템의 유연성을 높인다.



(그림 6) 유사 사용자들의 내역(History) 검사 모듈 구성

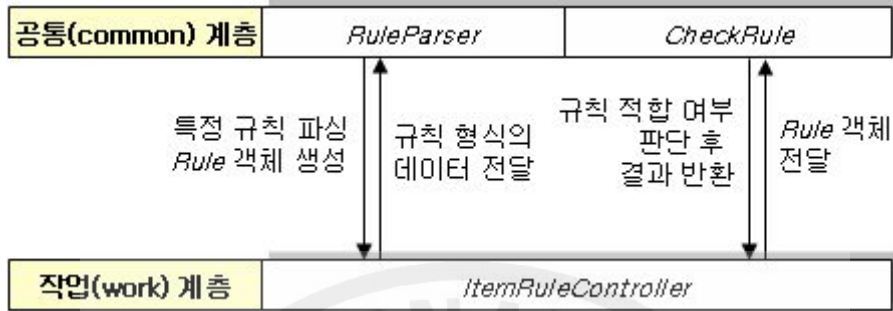
- 규칙기반 필터링 관련 모듈

본 연구에서 제안한 프레임워크는 아이템과 사용자 정보 사이에 특정한 규칙에 따라 추천이 가능하다. 그리고 필요한 경우에 특정 규칙을 설정해 놓음으로써 해당 규칙에 따라 아이템 추천의 우선순위를 조절한다. 본 논문에서 사용되는 규칙의 대표적인 구문 형식은 다음과 같다.

```
Class1:Field1='value' AND Class2:Field2='value' THEN Class:Field
```

이 구문의 의미는 *Class1*의 *Field1* 값이 *value*이고, *Class2*의 *Field2* 값이 *value*

이런 *Class*의 *Field*의 값을 오브젝트(Object) 형태로 반환한다. 이를 위해 규칙 생성과 규칙의 적합 여부를 판단하는 모듈들을 (그림7)과 같이 포함하고 있다.



(그림 7) 규칙 생성과 규칙 검사 모듈 구성

```

Rule Class Fields

public final static int AND =1;
public final static int OR =0;

private RuleUnit[] condition;
private int[] logicOpt;
private RuleUnit returnUnit;
    
```

(그림 8) *Rule* 클래스의 필드 정보

*Rule*의 구조는 (그림8)과 같이 조건부와 결과부 그리고 조건부에서 필요한 논리 연산(AND 또는 OR)들의 집합으로 구성된다. 각 부분은 또다시 작은 규칙 단위로 나누어진다. 그것이 바로 (그림9)에서 나타내고 있는 *RuleUnit*이다. *RuleUnit*은 *Rule*의 가장 작은 단위로써 클래스와 필드, 값 그리고 연산자로 나누어진다. 이 연산자는 '=', '!=', '<', '>' 중 하나를 취하게 되는데, 각각 'EQUAL'과 'NOT EQUAL', 'SMALL', 'BIG'을 의미하게 된다. *Rule*은 *RuleParser*를 통해 파싱이 이루어진다.

RuleUnit Class Fields

```
public final static int NOT_EQUAL=0;
public final static int EQUAL=1;
public final static int SMALL=2;
public final static int BIG=3;

private String cName; //클래스이름
private String fName; //field 이름
private String value; //field 값

private int opt; // '='는 0, '!'는 1,
// '<'는 2, '>'는 3을 의미한다.
```

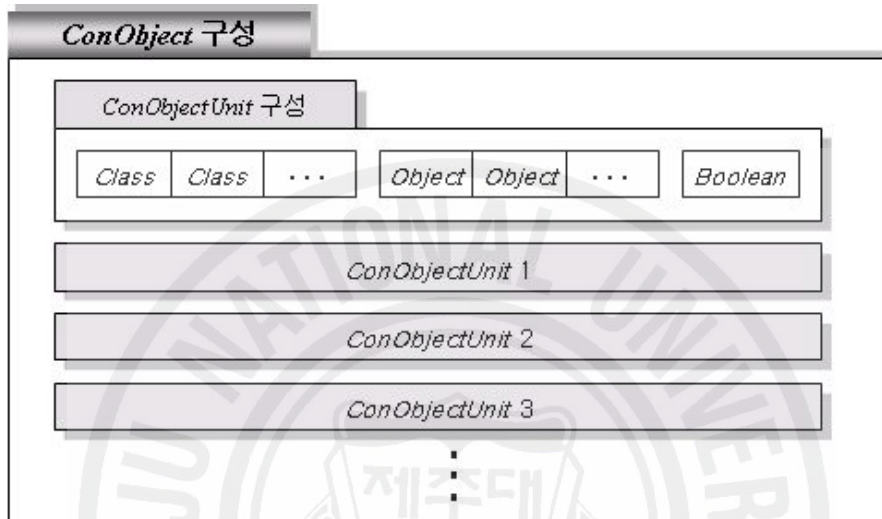
(그림 9) RuleUnit 클래스의 필드 정보

물론 Rule은 하나가 아닌 여러 개의 규칙들로 구성이 가능하다. 이는 ItemRuleController를 통해 순차적으로 적용된다. RuleParser를 통해 로딩된 규칙들은 앞서 언급한 ItemRuleController를 통해 다른 데이터들과 함께 CheckRule 모듈로 전달된다.

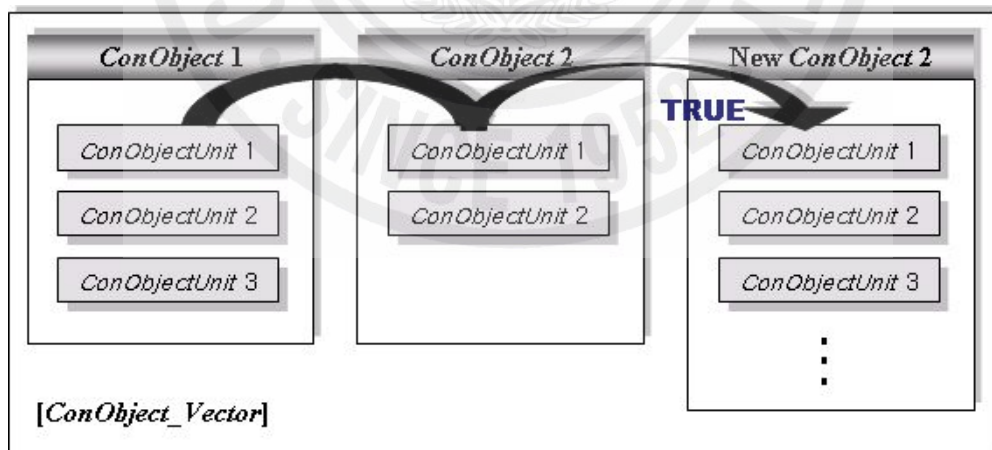
이후 CheckRule모듈은 ItemRuleController 객체를 오브젝트(Object) 타입의 상위 타입의 객체로 전달받는다. 이 객체를 가지고 메모리 상에 로딩된 값을 참고로 적합 여부를 검사한다. 이 객체로부터 추적된 클래스와 필드 정보 그리고 규칙 적합 여부를 결정하는 boolean 값들은 ConObject 단위의 Vector 객체에 저장된다. ConObject는 하나의 조건 규칙에 해당하는 단위이지만 그 내부에는 규칙에 설정된 클래스가 콜렉션(Collection) 타입의 클래스인 경우를 대비하여 ConObjectUnit으로 세부 단위를 둔다. ConObjectUnit은 (그림10)과 같이 클래스 타입의 벡터와 해당클래스 타입으로 활성화된 객체타입의 벡터, 그리고 규칙적합 여부를 판단하는 부울(boolean) 타입의 값으로 구성한다.

하나의 조건집합을 표현하는 ConObject와 또 다른 ConObject들 사이에 논리 연산을 통한 규칙 검사가 실행되면 (그림11)과 같이 ConObject 타입 벡터의 맨 마지막에 최종 규칙 적합 여부를 판단한 ConObject가 남게 된다. 맨 마지막 ConObject속에

*Rule*의 결과부와 일치하는 클래스와 필드가 존재한다면 찾아낸 필드의 값을 오브젝트 (Object) 형태로 *ItemRuleController*에 전달한다. 만약 *ConObject* 내에 일치하는 클래스와 필드가 없다면, *ItemRuleController* 객체 자체를 검색하여 로딩된 클래스와 필드들을 검사하여 일치하는 값을 돌려준다.



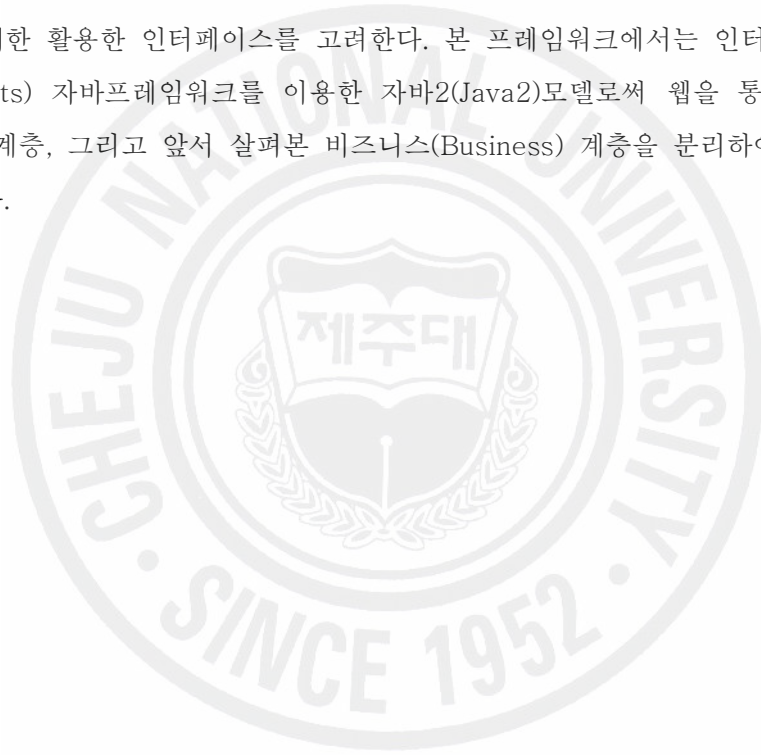
(그림 10) *ConObject*와 *ConObjectUnit*의 내부 구조



(그림 11) 조건부 검사 방법

3.2.3. 사용자 인터페이스(User Interface) 계층

사용자 인터페이스 계층을 통한 사용자 정보 입력은 사용자의 편의를 고려하여 2 단계로 입력을 받도록 하였다. 최초의 입력은 회원 가입 시기에 이루어지며, 이때 누락된 개인 선호 정보는 추후 사용자 정보 변경이나 아이템 선택을 통한 목시적 정보 수집을 통해 이루어진다. 두 번째 단계의 입력 시에는 추천에 필요한 새로운 선택 사항과 기존의 선택했던 내용들 중 원하는 항목을 선택하여 수정하도록 하였다. 많은 입력 요구는 사용자로 하여금 서비스를 회피하게 하는 요인이 되기 때문에 기존 입력 정보를 최대한 활용한 인터페이스를 고려한다. 본 프레임워크에서는 인터페이스를 스트럿츠(Struts) 자바프레임워크를 이용한 자바2(Java2)모델로써 웹을 통해 구현하여 뷰와 제어 계층, 그리고 앞서 살펴본 비즈니스(Business) 계층을 분리하여 독립성 있게 구성한다.



제 4 장. 프레임워크의 적용

본 장에서는 앞서 제안한 프레임워크를 이용하여 추천 시스템을 설계하고 구현한다. 이에 적용할 실제 아이템으로는 본 연구수행이 이루어진 대학교의 수강과목과 제주도내의 관광지로 선정하였다. 이러한 각각의 아이템에 대한 추천 시스템을 설계하고 구현한 사례를 보이고자 한다.

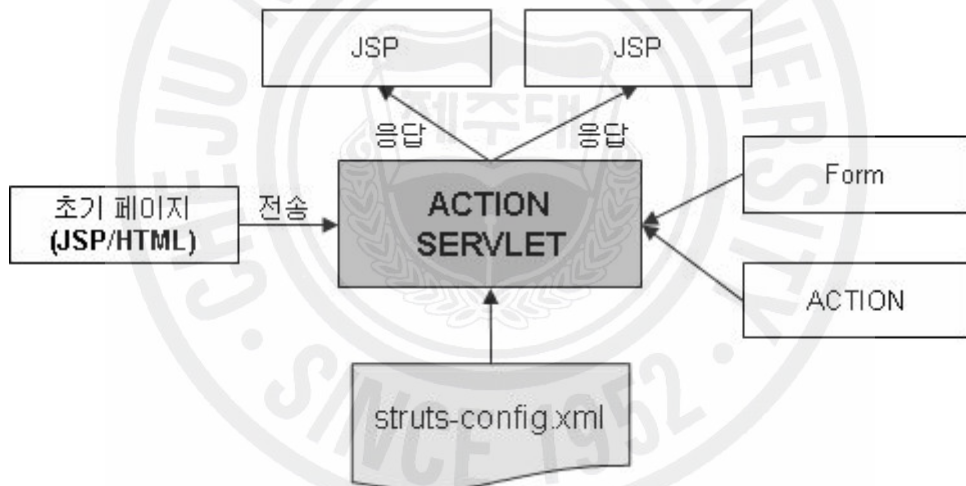
4.1 시스템 구현 환경

시스템의 구현에 이용한 시스템 환경은 <표1>과 같다. 주 프로그래밍 언어는 플랫폼에 독립적이고 확장성이 뛰어난 자바(Java)이고, 본 시스템은 개인화 서비스를 제공할 수 있는 웹 애플리케이션으로 제작하기 위해 스트럿츠(struts) 프레임워크를 기반으로 인터페이스를 구현한다.

<표 1> 시스템 구현 환경

편집 도구	Eclipse SDK 3.2.1	
구현 언어	Java, JDK1.5.0(Java Developer's Kit)	
웹 애플리케이션 프레임워크	Struts 1.1	
시스템 구성요소	웹서버	Apache-Tomcat5.5
	데이터베이스	MySQL Server 5.0
	운영체제	Windows Server 2003
	하드웨어	Intel(R) Xeon(TM) CPU 3.00GHz

스트럿츠(struts)는 자바빈즈(JavaBeans)나 자바 서블릿(Java servlets), 자바서버 페이지(JavaServer Pages, JSP) 등과 같은 표준 기술을 기반으로 만들어진 웹 애플리케이션 위한 프레임워크이다. 스트럿츠의 전체 동작 구조는 (그림12)과 같다. 스트럿츠는 HTTP요청에 대한 *ActionForm* 빈을 자동으로 설정해주기 때문에 웹 애플리케이션 개발자가 커스텀 코드를 직접 작성하고, 각 빈을 설정하는데 드는 노력과 시스템의 리소스 낭비를 줄여준다. 그리고 특정 모델에 종속되지 않고 스트럿츠 액션(Action)에서 헬퍼 클래스들을 호출하여 필요한 데이터에 접근하여 자바빈으로 가져온다. 이러한 자바빈을 통해 데이터에 접근하기 때문에 데이터 값을 나타내는 객체를 유지하기 쉬울 뿐만 아니라, 모델 계층에 대한 호출 횟수를 크게 줄일 수 있다는 장점이 있다. 이와 더불어 스트럿츠는 모델2(Model2) 기반의 뷰, 컨트롤러, 모델의 구조를 갖추고 있기 때문에 독립적인 확장성이 뛰어나다는 장점이 있다[21].



(그림 12) 스트럿츠의 전체 구조

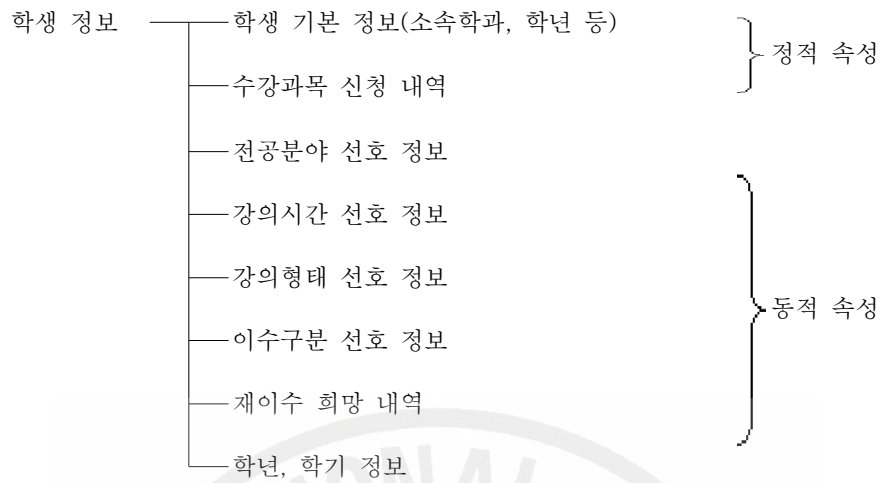
자료: T. Husted, C. Dumoulin, G.Franciscus, and D. Winterfeldt(2002), Struts 자바 프레임워크를 이용한 웹 애플리케이션의 구축(Struts In Action), 남기혁, 지상준 공역(2004), 인포북, p.40.

4.2 개인화된 수강과목 추천 시스템

4.2.1. 데이터 속성 구조

수강과목의 데이터는 제주대학교 개설과목을 대상으로 하였기에 제주대학교 수강편람을 기준으로 작성한다. 수강과목을 여러 기준과 특성에 따라 정확하게 분류하는 방법을 논하는 것은 교육학적으로 의미 있는 연구대상이지만 본 논문의 초점과는 거리가 있다. 따라서 비교적 객관적 기준으로만 데이터 속성들을 구성한다. 학생들이 수강과목을 선택할 때 기본적으로 고려하는 전공과 이수학점 그리고 선호하는 강의시간과 강의형태를 참고하여 구성한다. 따라서 데이터 선호 과목 속성 정보로는 전공, 이수학점, 강의시간으로 구성되고, 기본 속성은 과목명과 과목 식별자, 폐강유무와 대체과목 등을 포함한다. 강의시간은 수강편람에 그룹화된 수강과목 시간 그룹을 근거로 작성한다. 또한 학생 데이터는 이에 대응되는 선호속성들과 학년, 학기, 소속학과, 수강과목 내역 그리고 해당학기 이수학점과 재이수 신청과목을 속성으로 가진다.

위의 사항들을 고려하여 학생 속성들을 재구성하여 정리해보면 (그림13)와 같다. 사용자 입력의 최소화를 위해 사용자 데이터를 정적 속성의 데이터와 동적 속성의 데이터로 구분을 한다. 이 때 기준이 되는 것은 ‘개인이 얼마나 자주 필요한가?’에 따른다.



(그림 13) 재구성한 학생 속성

4.2.2 시스템의 설계

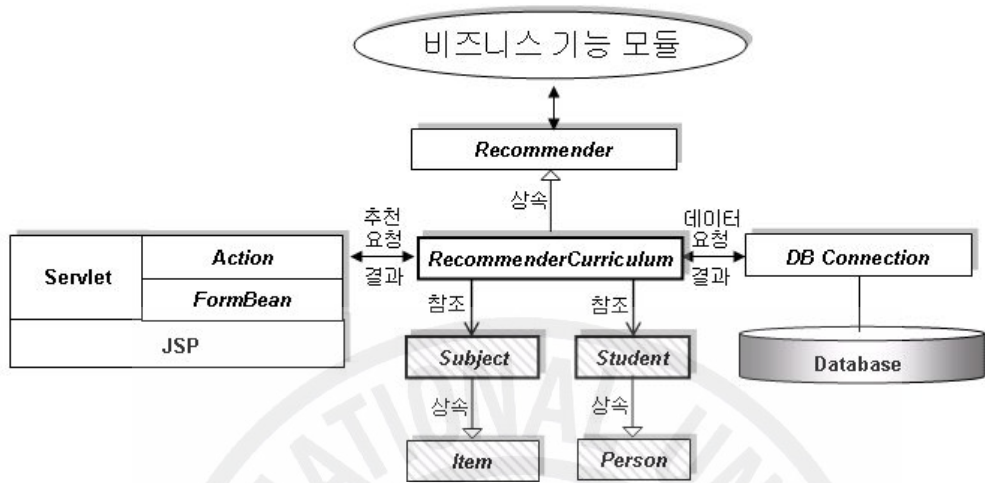
개인화된 수강과목 추천 시스템에서 필요한 기본 데이터는 학생과 과목에 대한 메타데이터 정보이다. 이를 위해 본 시스템은 각각의 클래스를 (그림14)와 같이 두 고 각 객체를 생성하여 프레임워크에 활용하였다.

<i>Student Class Fields</i>	<i>Subject Class Fields</i>
<pre> private int id; private String login_id; private String name; private String email; private String gender; private String birth; private String zipcode; private String addr1; private String addr2; public int grade; //학년 public int term; //학기 public String depart; //소속학과 public String student_id; //학번 private int major[]; //선호전공분야 private int course[]; //중요이수구분 private int lecture[]; //선호강의형태 private int time[]; //선호강의시간 private int history[]; //수강과목 내역 private int deptId; //소속학과 식별자 public int study_point; //해당학기 이수학점 public int[] re_study; //재이수 과목 id </pre>	<pre> private String name; //과목명 private int id; //과목 식별자 private int major; //개설학과 일련번호 private int course; //이수영역구분 private int lecture; //강의형태 private int time; //배정시간 그룹 일련번호 private String pf_name; //과목 담당 교수 public int point; //할당된 이수학점 public int grade; //학년 public int term; //학기 public int[] before; //선수과목 id public int isOpen; //개설:1, 폐강:0 public int[] replace; //대체과목 id </pre>

(그림 14) *Student*와 *Subject* 클래스의 필드(Fields) 구조

전체적인 시스템 구조는 (그림15)와 같다. 3장에서 언급한 바와 같이 *Recommender* 모듈의 자식인 *RecommenderCurriculum* 모듈은 비즈니스 계층의 3가지 필터링을 제어하는 역할을 한다. 각 필터링이 하는 역할을 수강과목 추천 과정을 통해 살펴보면 다음과 같다.

개인화된 수강과목 추천 시스템 구조

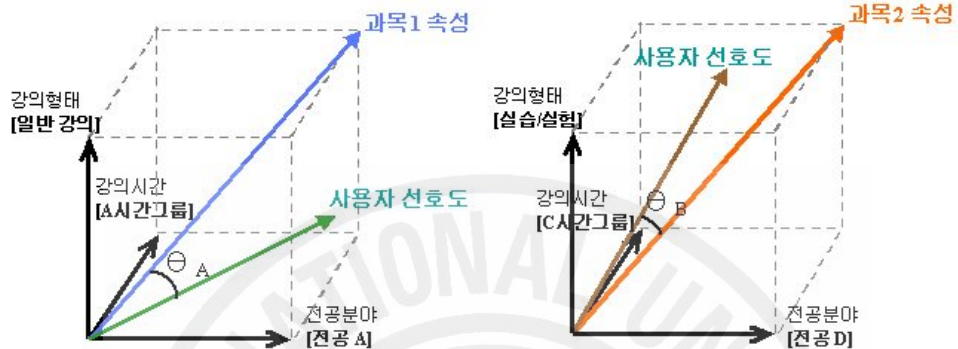


(그림 15) 개인화된 수강과목 추천 시스템 구조도

첫째, 내용기반 필터링 관련하여 유사도 계산식을 적용한 예이다. (그림16)의 속성 정보와 같은 과목 A, B와 개인 사용자 선호도가 있다고 가정할 때, 수강과목이 갖는 속성 값들을 최상의 선호 값으로 두고 사용자의 선호도가 수강과목 속성과 얼마나 유사한가를 측정한다.

(그림16)의 상단에 있는 ‘수강과목 정보’는 전공분야가 전공A이고, 강의형태가 일반 강의이고, 강의시간이 A시간그룹인 과목1과 전공분야가 전공D이고, 강의형태가 실습/실험이며, 강의시간이 C시간그룹인 과목2의 특성정보를 나타낸다. 그리고 ‘사용자 선호 정보’를 갖는 개인사용자가 존재할 때 위의 두 개의 그림은 각각 과목1과 개인 사용자 그리고 과목2와 개인사용자 사이의 유사성을 시각화한 것이다. 과목1의 특성을 최대값으로 두고, 최대값을 갖는 측과 관련된 사용자 속성 값들을 근거로 유사도를 계산한다. 따라서 과목1(전공A, 일반 강의, A시간그룹)에 해당하는 개인사용자의 선호도 값(전공A: 3, 일반 강의: 2, A시간그룹: 4)을 가지고 코사인 값을 측정하고, 과목 B(전공D, 실습/실험, C시간그룹)에 해당하는 개인사용자 선호도 값(전공D: 2, 실습/실험:3, C시간그룹: 2)을 가지고 코사인 값을 측정하여 최대 유사도를 갖는 과목을 추천한다.

수강과목 정보			사용자 선호 정보				
전공 분야	과목 1 전공 A	과목 2 전공 D	전공 분야	전공 A	전공 B	전공 C	전공 D
강의 형태	일반 강의	실습/실험	우선순위	3	4	1	2
강의 시간	A 시간그룹	C 시간그룹	강의 형태	일반 강의	토론/토의	프로젝트	실습/실험
			우선순위	2	4	1	3
			강의 시간	A 시간그룹	B 시간그룹	C 시간그룹	D 시간그룹
			우선순위	4	3	2	1



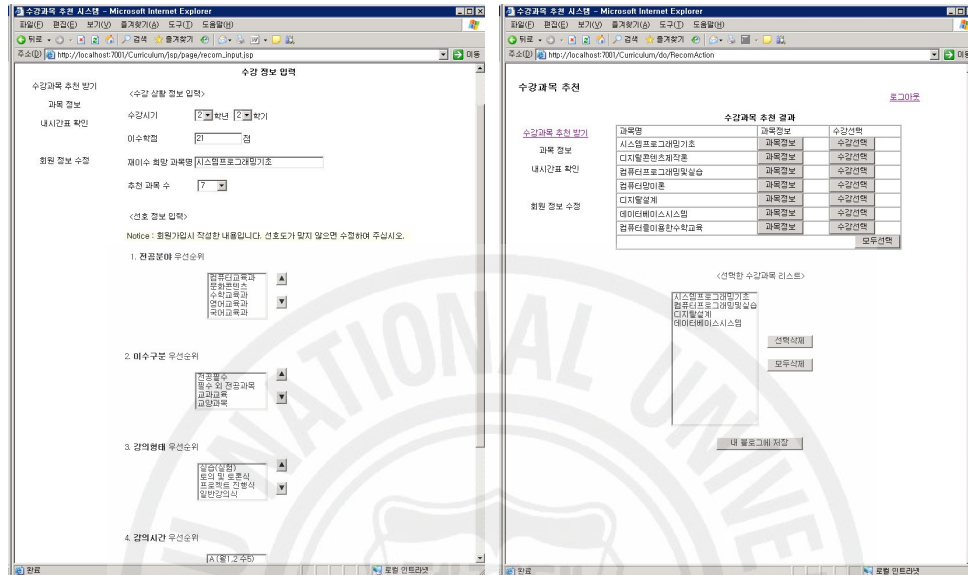
(그림 16) 수강과목에 대한 내용기반 필터링의 적용

둘째, 수강신청 내역의 유사 사용자들을 고려하여 개인 사용자에게 수강과목 추천한다. 즉, 기존 사용자들의 수강과목 내역들을 토대로 수강 패턴을 조사하여 그 결과를 가지고 추천에 반영한다. 셋째, 규칙기반에 있어서는 아래와 같이 관련 과목의 우선순위를 조절하는 규칙을 이용한다.

```
Subject:major='컴퓨터교육과' AND Subject:grade='2'
AND Subject:term='2' THEN Subject:id";
```

이 규칙에 들어가는 내용은 실제 구현상으로는 본래 전공 식별자와 사용자 관련 정보 변수가 값으로 들어가지만 편의상 여기서는 문자열로 대체한다. 규칙의 사용 예는 컴퓨터 교육과 2학년 학생이 2학기 수강신청을 할 경우, 해당과목의 개설학과가 컴퓨터교육과이고, 해당과목이 2학년 2학기 권장과목이면 해당 과목의 식별자를 반환하라는 우선순위 조절이 가능하다.

4.2.3. 시스템의 구현



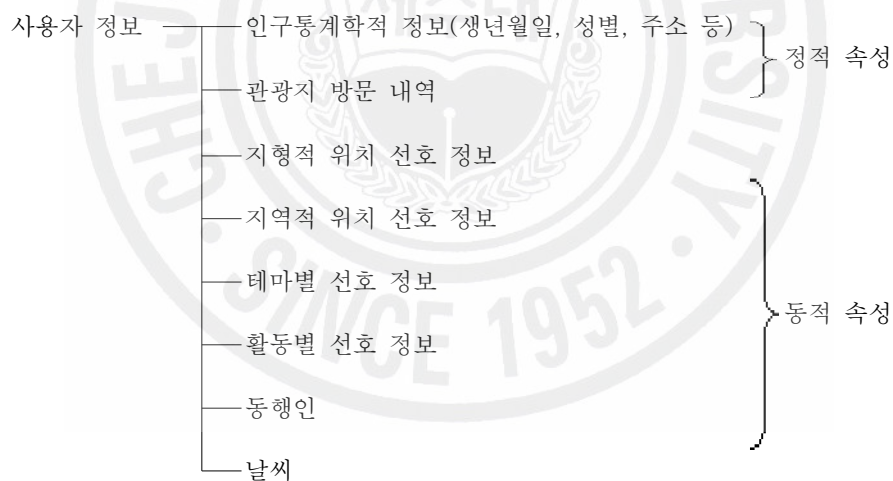
(그림 17) 수강과목 추천을 위한 입력과 출력 화면

수강과목 추천 시스템의 회원가입 후 사용자의 기본 정보와 과목 수강 신청 시 특이사항을 입력하도록 함으로써 추천의 토대 정보를 구축한다. 수강신청이 이루어질 대상 학기와 총 이수학점 그리고 재이수가 필요할 경우 재이수할 과목명을 입력한다. 한 과목당 이수 학점이 대부분 3학점인 점을 감안하고, 한 학생이 한 학기에 이수할 수 있는 총 학점이 20점 내외인 것을 고려하여 기본 추천 과목의 수는 7과목으로 지정하였으나 사용자가 수정 가능하다. 또한 회원 가입할 당시에 등록했던 선호 정보들을 자신이 원하는 우선순위에 따라 선택적으로 재조정이 가능하다. 이러한 정보들을 통해 (그림17)과 같이 수강과목 추천이 이루어지면 리스트에서 원하는 과목을 지정하여 시간표를 구성한다.

4.3 개인화된 관광지 추천 시스템

4.3.1. 데이터 속성 구조

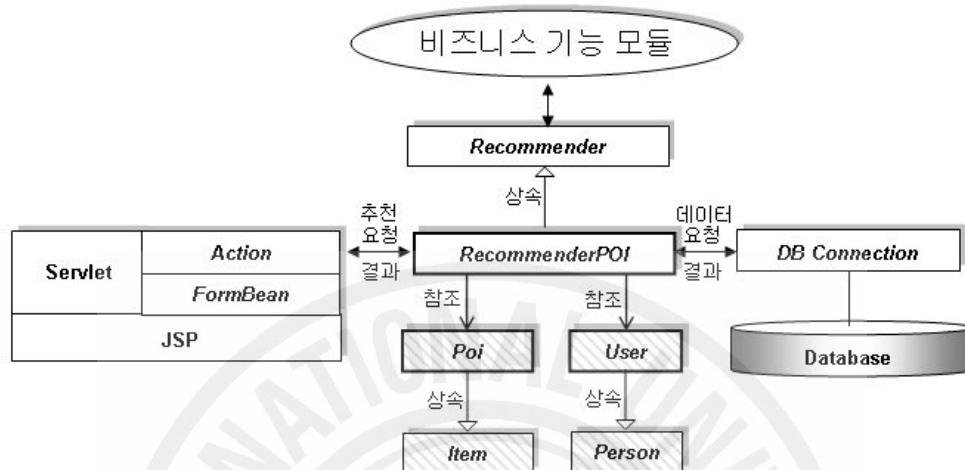
개인화된 관광지 추천을 위해서는 2가지 기본적인 정보, 관광지 자체에 대한 정보와 방문자(개인 사용자)에 대한 정보가 필요하다. 본 연구에서는 제주도 관광을 안내하는 공공기관의 사이트들의 관광지 소개와 관련된 디렉터리 구조를 재구성하여 메타데이터를 구성한다. 따라서 추천 시스템에 사용된 관광지 정보는 객관적 속성들에 중점을 둔다. 객관적 속성은 관광지가 위치한 지리적 위치와 같이 누가 분류하더라도 일반적인 동의를 얻을 수 있는 관광지 정보들을 말한다[27]. 크게 지형적 위치와 지역적 위치 그리고 테마별, 활동별, 성수기 특성을 가지고 관광지 정보들을 구축한다. 사용자 속성들을 재구성하여 정리하면 (그림18)과 같다.



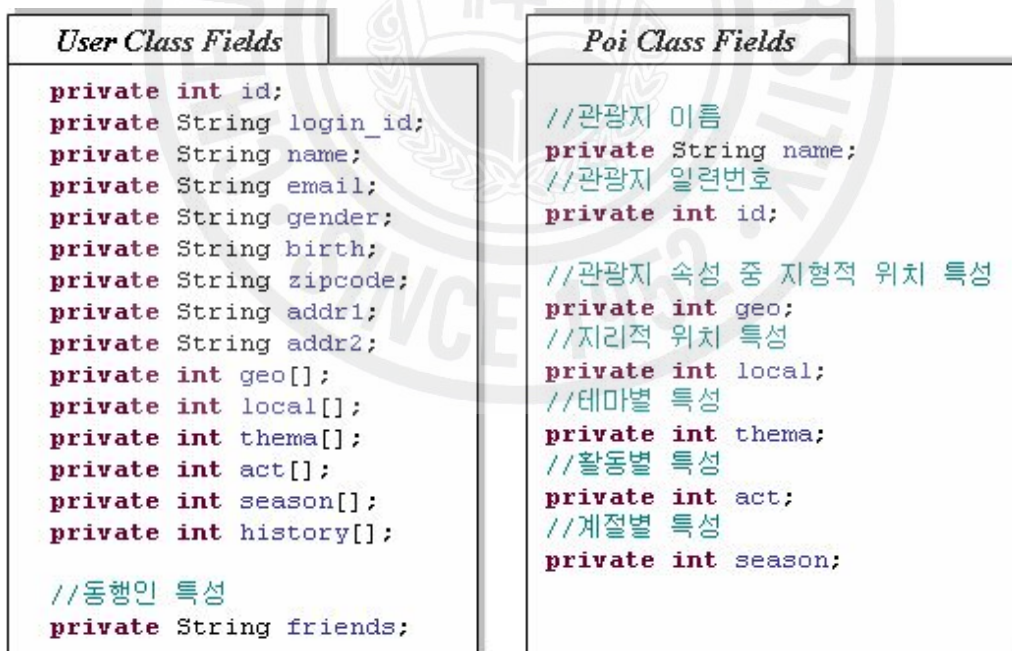
(그림 18) 재구성한 사용자 속성

4.3.2. 시스템의 설계

개인화된 관광지 추천 시스템 구조



(그림 19) 개인화된 관광지 추천 시스템 구조



(그림 20) Poi와 User 클래스의 필드(Field) 구조

개인화된 관광지 추천 시스템의 전체적인 구조는 (그림19)와 같다. (그림20)은 Poi와 User의 클래스 구조를 나타낸다. 비즈니스 계층의 주요 모듈들은 반드시 Poi와 User로 생성된 객체들을 이용하여 연산을 수행한다. 세 가지 필터링을 활용하는 방법들을 살펴보면 다음과 같다.

첫째, 내용기반 관련 모듈에서는 관광지 속성과 사용자 선호도 간의 유사성을 계산한다. 둘째, 관광의 경우 유사한 동선을 가지고 관광지를 관광하게 되는 사용자들이 존재한다. 이에 착안하여, 개인 사용자가 관광지A->관광지B->관광지C의 순서로 관광을 하였고, 다른 사용자가 관광지A->관광지B->관광지C->관광지D의 순서로 관광을 하였다면 둘의 동선이 비슷하게 진행된다는 것을 의미한다. 따라서 개인 사용자와 다른 사용자 사이의 내역을 비교하여 동일한 관광지를 선택하고, 동일한 순서로 관광지를 선택한 패턴이 발견된다면 타사용자가 방문했지만 개인사용자는 방문하지 않은 관광지D를 우선적으로 추천받는다. 셋째, 규칙에 기반을 둔 필터링에 사용가능한 규칙의 예를 들어보면 다음과 같다.

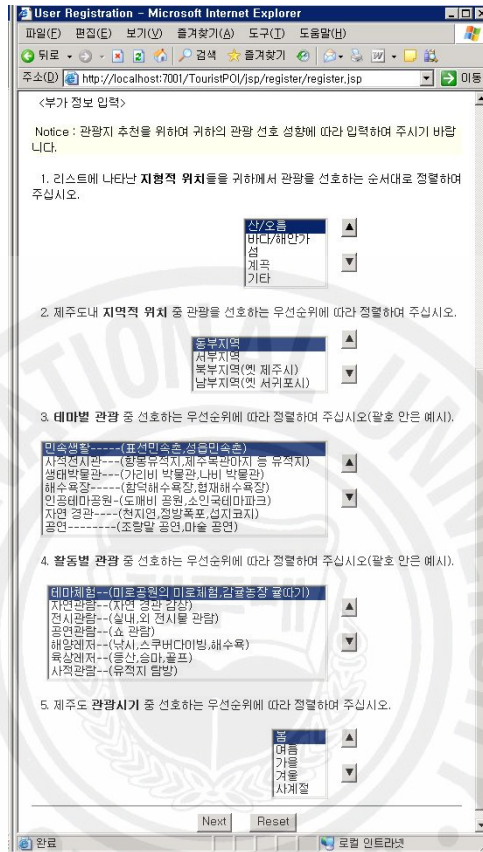
- (a) User:friends='family_young' AND Poi:thema='인공테마공원'
THEN Poi:id
- (b) Weather:temp>'23' AND Weather:wind<'5' AND
Weather:rain<'5' AND Poi:act='해양레저' THEN Poi:id

(a)는 사용자의 동행인이 미성년자를 동반한 가족인 경우 테마공원(놀이공원 또는 미로체험)의 우선순위를 높이는 규칙이다. (b)규칙에서는 하루 평균온도가 23도 이상이고, 바람이 5m/s 미만이며 강수량이 5mm이하인 경우 해양레저를 추천하는 규칙을 의미한다. 이와 같이 특정 날씨에 관하여 특정한 관광지를 추천하는 규칙도 가능하다.

4.3.3. 시스템의 구현

(그림21)은 시스템이 구현된 모습을 보여준다. 회원 가입 단계에서 기본 정보를

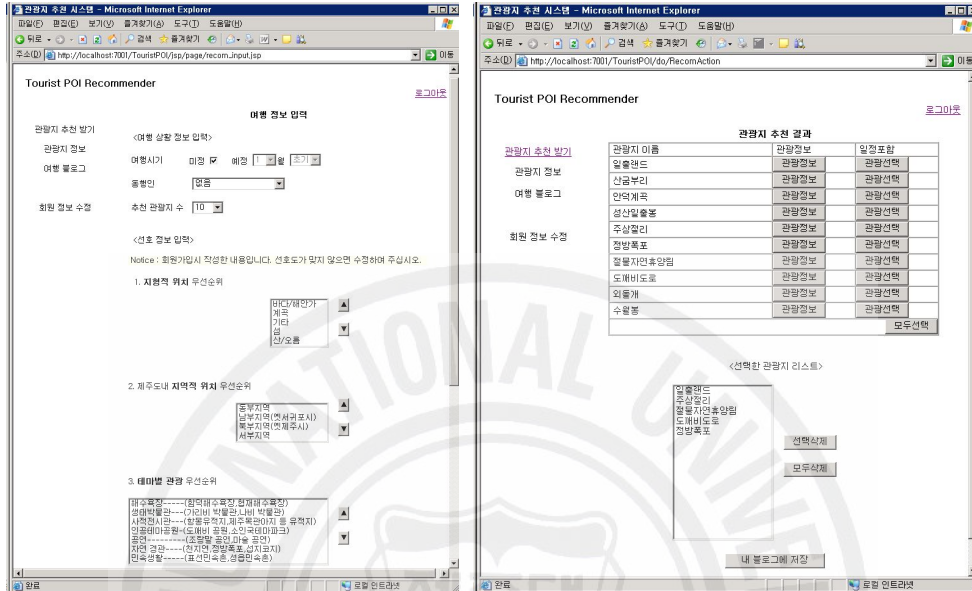
입력 받는다. 이 때 관광 속성에 대한 선호 정보는 우선순위를 변경하며 입력받게 된다.



(그림 21) 회원 등록 중 관광지 선호 정보 입력

추천 받기의 입력 단계에서는 변경이 필요한 정보만을 입력하면 된다. 본인에 대한 기존의 관광지 선호 정보도 함께 나타나 있으므로, 만약 변경하고 싶을 경우에는 우선순위를 조절한다. (그림22)는 개인 사용자의 로그인 후 추천 받기의 입력화면이다. 회원 가입 시와는 다르게 본인의 기존의 관광지 선호 특성이 미리 세팅되어 있는 것을 알 수 있다. 여행 날짜는 날씨 DB를 통해 특정 시기의 레저관광의 안전을 위해 입력하는 것이므로 관광지의 성수시기와 맞물리게 되는 사용자 선호도로 입력되는 계

절과는 다른 의미로 사용된다. 이곳에서 사용자가 원하는 추천 관광지의 수와 동행인의 정보가 시스템에 입력된다.



(그림 22) 관광지 추천을 위한 입력과 출력 화면

시스템에 의해 추천된 관광지는 (그림22)과 같이 리스트 형식으로 보인다. 이후 개인사용자는 추천 관광지 중 자신이 원하는 관광지를 선택하여 아래의 선택상자 안에 담게 되고, 선택상자 안의 관광지들은 개인 기록함에 저장된다.

제 5 장. 성능 평가

5.1 수강과목 추천 시스템 성능 평가

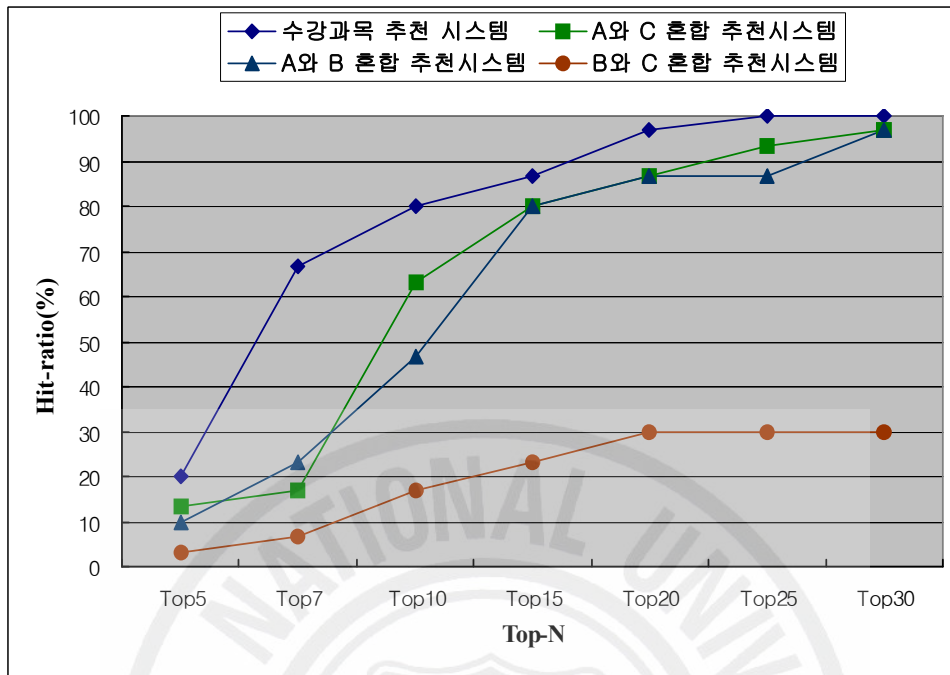
본 시스템은 제주대학교 컴퓨터교육과 학생들의 수강과목을 토대로 하여 총 126개 과목과 30명의 수강생을 대상으로 추천을 실시한다.

$$\text{적중률}(\text{Hit-ratio}) = \frac{\text{Number of Hits}}{\text{Number of Events}} \quad (3)$$

본 시스템의 성능의 우수성을 확인하기 위해 본 시스템과 추천의 방법을 달리한 시스템들의 추천 적중률을 식 (3)을 이용하여 비교하였다. 추천의 방법은 다음과 같다.

- (A) : 사용자 선호 정보와 아이템 정보 사이의 유사도 계산을 이용한 추천
- (B) : 사용자 내역과 선호 정보를 토대로 하여 협동적 필터링을 통한 추천
- (C) : 설정한 규칙에 의한 추천 방법

이 추천 방법들을 2개씩 혼합한 추천 방식을 적용한 시스템들을 비교 대상으로 설정한다. 이는 앞서 언급한 바와 같이 단일 필터링을 사용하는 것보다 2개의 혼합필터링을 사용하는 것이 유리하기 때문이다. 적중률의 적중 기준은 평균적으로 실제 선택한 수강과목들의 50%이상을 맞췄을 경우 적중한 것으로 간주한다. 이에 다음과 같은 결과가 나온다.



(그림 23) 수강과목 추천 시스템 성능 평가

<표 2> 수강과목 추천 시스템의 적중률

추천 수	적중률(Hit-ratio, %)						
	Top5	Top7	Top10	Top15	Top20	Top25	Top30
수강과목 추천 시스템	20.0	66.7	80.0	86.7	96.7	100.0	100.0
A와 C 혼합 추천 시스템	13.3	16.7	63.3	80.0	86.7	93.3	96.7
A와 B 혼합 추천 시스템	10.0	23.3	46.7	80.0	86.7	86.7	96.7
B와 C 혼합 추천 시스템	3.3	6.7	16.7	23.3	30.0	30.0	30.0

(그림23)과 <표2>에서 나타난 바와 같다. Top-N은 추천에 사용된 아이템 중 추천 우선순위가 가장 높은 것부터 시작하여 N개를 뽑은 것을 의미한다. Top-5에서 적중률이 지나치게 낮은 이유는 적중된 과목의 수를 목표 아이템의 50%이상 맞추도록 하였으나 목표 아이템의 50%와 Top-5개의 추천 수가 근접하여 적중률이 상대적으로 낮았다. 그러나 이러한 가운데서도 타 시스템에 비해 본 연구에서 제안한 모든

모듈을 혼합한 시스템이 제일 우수한 적중률을 보이고 있다. 결과적으로 추천의 방법을 달리한 3개의 시스템들의 추천 적중률보다 본 시스템이 전체적으로 높은 적중률을 보인다. 즉, A, B, C 각각의 방식을 제외한 추천 시스템에서는 3가지를 혼합한 수강과목 추천 시스템보다 전체적으로 낮은 적중률을 보이고 있다. 이는 본 시스템에서 사용한 추천 기법들이 각각 추천에서 적중률을 높이는 데 중요한 역할을 담당하고 있음과 동시에 이것이 통합된 시스템인 본 시스템에서 가장 좋은 성능을 가지고 있음을 보여준다.

추천 시스템에서 적중률과 함께 고려될 수 있는 부분은 바로 재현율(recall-ratio)과 정확도(precision-ratio)이다. 재현율과 정확도는 본래 정보 검색분야에서 검색의 효율을 검증하기 위해 일반적으로 많이 사용되는 방법이지만, 최근에는 추천 시스템의 성능 평가에도 쓰인다. 재현율은 사용자가 원하는 아이템으로 잡은 목표 아이템들에 대한 목표 아이템 내의 추천 아이템들의 적중 비율이다. 재현율의 계산식은 식 (4)와 같다. 그리고 정확도는 추천 아이템들에 대한 목표 아이템 내의 추천 아이템들의 적중 비율이다. 정확도의 계산식은 식(5)와 같다.

$$\text{재현율 (Recall-ratio)} = \frac{\text{Number of Hit_Items}}{\text{Number of Target_Items}} \quad (4)$$

$$\text{정확도 (Precision-ratio)} = \frac{\text{Number of Hit_Items}}{\text{Number of Recommended_Items}} \quad (5)$$

즉, 목표 아이템이 {a, b, c, d}이고, 추천아이템이 {a, c, e, f, g}라면 재현율은 50%가 된다. 그리고 정확도는 40%가 된다. <표3>은 수강과목 추천 시스템의 평균 재현율을 나타낸 표이다. 적중률과 마찬가지로 모든 모듈을 혼합한 수강과목 추천 시스템이 전체적으로 높은 재현율을 나타냈다.

<표 3> 수강과목 추천 시스템의 평균 재현율

추천 수	재현율(Recall-ratio, %)					
	Top5	Top10	Top15	Top20	Top25	Top30
수강과목 추천 시스템	45.68	62.19	71.25	78.37	79.51	83.50
A와 C 혼합 추천 시스템	40.78	58.07	70.14	73.25	77.70	81.20
A와 B 혼합 추천 시스템	33.37	41.49	51.96	61.83	69.45	79.12
B와 C 혼합 추천 시스템	23.08	31.95	35.70	37.92	37.92	38.40

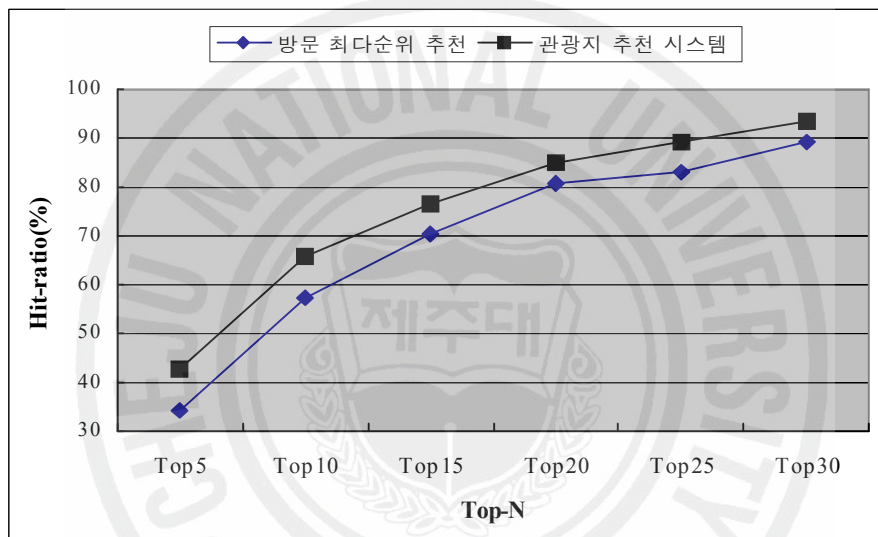
<표 4> 수강과목 추천 시스템의 평균 정확도

추천 수	정확도(Precision-ratio, %)					
	Top5	Top10	Top15	Top20	Top25	Top30
수강과목 추천 시스템	43.33	29.66	22.88	19.00	15.46	13.66
A와 C 혼합 추천 시스템	36.66	27.33	22.22	17.50	15.06	13.00
A와 B 혼합 추천 시스템	31.33	19.66	16.66	14.83	13.46	12.88
B와 C 혼합 추천 시스템	23.61	17.30	13.07	10.38	8.30	7.05

<표4>는 수강과목 추천 시스템의 평균 정확도를 나타냈다. 여기서 평균 정확도는 추천 개수가 늘어날수록 감소하는 경향을 보인다. 이는 목표 아이템의 수가 평균적으로 5개로 추천 수보다 작기 때문에 Top-N에서의 N값의 증가량만큼 목표아이템이 적중률이 따라가지 못하는데 그 원인이 있다. 또한 앞서 언급한 바와 같이 실험에 사용한 목표 아이템의 수가 N보다 작게 제한되어 있어 최고의 적중을 한다고 가정하여도 계산식 상으로 대부분 100%에 미치지 못한다. 따라서 이를 절대적인 정확도로 파악하기에는 다소 무리가 있다. 따라서 본 연구에서는 정확도를 타 시스템과의 비교를 통해 상대적인 의미로 분석한다. 이와 같은 관점에서 볼 때, <표4>에서 알 수 있듯이 이 역시 A, B, C 모두 혼합된 추천 시스템의 정확도가 다른 것에 비해 높게 유지되고 있다.

5.2 관광지 추천 시스템 성능 평가

제주도 관광 협회에 등록된 관광지과 해수욕장 등을 기준으로 96개의 관광지과 관광객 47명을 대상으로 본 시스템의 성능 평가를 수행하였다. 실험의 방법과 성능 평가에 사용된 공식은 앞서 살펴본 5.1장의 내용과 동일하다. 그리고 본 연구에서는 추천한 관광지과 사용자가 최근에 선택한 관광지 간의 적중률(hit-ratio)을 측정하였다.



(그림 24) 관광지 추천 시스템의 성능 평가

<표 5> 관광지 추천 시스템의 적중률

추천 수	적중률(Hit-ratio, %)					
	Top5	Top10	Top15	Top20	Top25	Top30
방문 최다 순위 추천	34.0	57.4	70.2	80.9	83.0	89.4
관광지 추천 시스템	42.6	66.0	76.6	85.1	89.4	93.6

그 결과 방문 최다 순위에 따른 추천 방식보다 본 연구에서 개발한 시스템의 성능

이 (그림24)와 <표5>에 나타난 바와 같이 평균적으로 7%정도 더 우수하였다. 이로써 본 시스템은 관광지 방문횟수에 기초한 유명 관광지 추천보다 뛰어난 성능을 나타낼 수 있었다.

<표 6> 관광지 추천 시스템의 평균 재현율

추천 수	재현율(Recall-ratio, %)					
	Top5	Top10	Top15	Top20	Top25	Top30
방문 최다 순위 추천	8.5	15.36	27.88	40.10	40.87	45.39
관광지 추천 시스템	14.9	27.53	34.93	41.02	45.77	52.38

<표 7> 관광지 추천 시스템의 평균 정확도

추천 수	정확도(Precision-ratio, %)					
	Top5	Top10	Top15	Top20	Top25	Top30
방문 최다 순위 추천	9.09	8.63	10	10.79	8.9	8.33
관광지 추천 시스템	14.54	13.63	11.96	11.96	9.54	9.16

<표6>과 <표7>은 관광지 추천 시스템에 대한 평균 재현율과 평균 정확도를 나타낸다. 두 검증 역시 본 연구에서 제안한 시스템이 전체적으로 우수한 성능을 보이고 있음을 나타낸다. 단, <표7>에서 방문 최다 순위 추천의 정확도가 Top10에서 Top20 구간에서 줄지 않고 증가하는 경향을 볼 수 있는데, 이는 <표6>에서 알 수 있듯이 다른 구간보다 재현율의 증가가 급격하게 증가한데 기인한다. 즉, 추천한 아이템이 갑자기 목표 아이템 집합에 많이 있었던 결과 나타난 일시적인 현상이다. 그러나 비율 수치상으로 보아도 본 연구가 제안한 시스템이 더 높은 정확도를 보이고 있다.

5.3 프레임워크 성능 분석

앞서 5.1과 5.2절에서 본 연구의 프레임워크를 적용한 전혀 다른 분야의 두 주제

를 가지고 추천 시스템을 구현하고 이들의 추천 적중률을 측정하였다. 그 결과 타 시스템보다 높은 적중률을 갖고 있음을 보였다. 즉, 아이템 특성과 추천 방법이 다른 두 주제를 가지고 동일한 프레임워크로 우수한 추천 시스템을 구현하였다. 프레임워크의 성능은 주로 생산성과 품질을 기준으로 많은 개발자들과 적용 경험을 통해 평가되어지는 것이 가장 바람직한 평가방법이지만, 본 연구와 같이 단기간의 소규모 프레임워크 개발에 있어 많은 개발자들의 적용 개발 결과로 성능을 평가하기는 어렵다. 기존의 프레임워크 관련 논문들 역시 이러한 문제로 인하여 프레임워크를 구현한 후 해당 프레임워크의 적용에 대한 성능평가로 분석을 하거나, 내부 모듈의 성능평가를 한다. 본 연구에서는 이에 추가적으로 프레임워크의 구조적인 분석을 통해 프레임워크의 생산성을 보인다. 이를 위해 본 연구의 생산성을 높이는 구조적인 특성을 보이고, 코딩 라인 수와 모듈 수를 이용하여 생산성을 분석한다. 프레임워크의 구조적인 특성을 분석하기 위한 기준으로 앞서 3장에서 제시한 프레임워크의 요구분석을 토대로 한다.

첫째, 데이터 속성 구조와 데이터 추천 처리 모듈을 분리하였다. 본 연구에서는 추천을 하고자 하는 모든 것(물품, 웹기반 콘텐츠, 웹 문서 등)을 총칭하여 *Item*이라는 객체로 생성하고, 추천을 받는 대상을 *Person*으로 생성하였다. 그리고 추천 처리에 필요한 각 객체의 식별자와 속성내용을 등록할 수 있는 해시테이블을 두어 모든 아이템과 사람의 정보를 관리하였다. 그러나 각 아이템은 고유의 속성을 가지고 있다. 따라서 상속을 통해 각 아이템과 사용자들의 특성을 반영하여 확장할 수 있는 방법을 제시하였다. 추천 정보를 처리하는 모듈에서는 이러한 상위 객체 정보만을 가지고 추천을 진행하였다. 따라서 어떠한 특징을 가진 아이템이 온다고 하더라도 추천의 처리는 독립적으로 이루어진다.

둘째, 추천을 처리하는 기술이 확장된 새로운 모듈 수용이 가능하다. 본 논문에서는 필터링 기술을 혼합할 수 있는 시스템을 구현하고 있다. 상황에 따라 본 연구에서 기본적으로 모듈 단위로 필터링을 제시하고 있고, 제안한 모듈 외에 더 필요한 모듈이 있다면 추천을 처리하는데 있어 제어 모듈을 통해 새로운 필터링 모듈을 추가해 나갈 수 있도록 하였다.

셋째, 추천 대상에 따라 추천을 처리하는 과정이 다양하게 제어되도록 고려하였다. 본 논문에서는 한 가지 추천 흐름만을 제시하는 정적인 추천이 아니라 제어 모듈의

확장을 통해 동적인 추천이 가능하다. 즉, 모듈 단위의 추천 기술들을 조합하여 다양한 방식의 추천 흐름과 우선순위 및 가중치 조절 등을 통해 아이템 특성에 맞는 제어 모듈을 구성하도록 하였다. 이와 같이 데이터 속성 구조 정의와 추천 제어 모듈만의 구성으로 추천시스템을 구현할 수 있는 프레임워크를 제안함으로써 우수한 추천시스템을 손쉽게 구축가능하게 하였다.

이러한 특징을 바탕으로 생산성의 정량적 측정을 위하여 단일 목적으로 개발된 관광지 추천 시스템[27]의 개선된 프로그램과 본 연구의 프레임워크에 추가적으로 관광지 추천을 적용한 사례를 가지고 비교하고자 한다. 코딩 라인 수에 포함시킨 모듈들은 각 응용에서 실제 개발한 핵심 모듈로 제한한다. 그리고 이에 포함되는 모듈들은 데이터 속성 구조 객체, 사용자 속성 구조 객체, 추천 전략 관련 모듈이다. 이들의 코딩 라인과 모듈 수는 <표8>과 같다. 여기서는 앞서 언급한 단일 목적으로 개발된 추천 시스템을 A 시스템이라 하고, 프레임워크를 기반으로 적용한 시스템을 B 시스템이라 한다.

<표 8> 프로그램 코드 개발 생산성 비교

※ ()안: 전체 시스템 코딩 중 실제 구현에 필요한 모듈의 코딩 라인 수의 비율

분석 대상	A 시스템	B 시스템
라인(line) 수	2,117(100%)	854(34.16%)
모듈 수 (실제 구현 / 전체 추천 모듈)	16/16(100%)	4/20(25%)

두 시스템의 추천 방법적 측면에서의 알고리즘과 적중률은 비슷하지만 <표8>에서 보이는 바와 같이 두 시스템 사이의 생산성을 비교해 보면 B 시스템이 A 시스템보다 약 60%정도 높다. 이는 개발자가 추천 시스템을 구현하고자 할 때 A 시스템은 필수 모듈의 100%를 모두 개발해야 하지만 B 시스템에서는 계층적인 구조를 취함으로써 중복을 최대한 줄여 전체 코딩의 약 34%, 전체 모듈의 약 25%만을 구현하면 되기 때문이다. 따라서 본 프레임워크를 이용하여 추천 시스템을 개발할 경우 코드 개발 비용 및 시간 비용을 절감하여 생산성을 향상시킬 수 있다.

제 6 장. 결 론

본 논문에서 제안한 하이브리드 필터링 기반 개인화된 추천 프레임워크를 통해 수행한 내용은 다음과 같다.

첫째, 추천 방법에 따른 개인화된 서비스 제공 기술을 통합적으로 응용한 프레임워크를 개발한다.

둘째, 개인화를 위한 데이터 속성을 연구하고 본 연구의 프레임워크를 적용하여 실제 추천 서비스 응용 시스템을 구현한다.

셋째, 하이브리드 방식의 추천 기술을 적용한 프레임워크의 성능을 평가하여 타 추천 방법보다 우수한 성능을 나타냄을 보인다.

본 연구에서는 데이터의 속성들을 구조화하고, 정보필터링 추천 기술들을 응용한 하이브리드 필터링 기술을 연구하여 프레임워크를 설계하였다. 그리고 제안한 프레임워크를 이용하여 제어 모듈과 데이터 속성 객체의 추가만으로 추천 시스템들을 구현하였다. 그리고 그 결과 본 논문이 제안한 프레임워크를 적용한 시스템들은 우수한 성능을 보였다. 이는 본 연구에서 제안한 프레임워크를 통해서 다양한 분야에 적용하여 효율적이고 우수한 추천 시스템을 구현할 수 있음을 의미한다.

향후 본 프레임워크의 사용자 인터페이스를 보완하여 다양한 형태의 기기에 적용이 가능한 사용자 인터페이스 구축이 요구된다. 또한 규칙 기반 모듈을 응용, 발전시켜 지능적인 추론을 통한 추천이 가능하도록 하는 연구가 필요하다. 그리고 다양한 추천 대상 아이템들에 대한 특성들의 충분한 고려 속에서 적절한 데이터 속성 구조를 빠르며 손쉽게 구축할 수 있는 방법론에 대해서도 추후 논의가 필요하다.

참 고 문 헌

- [1] 백장현, 김영식(2003), 웹 기반 교육 환경에서 만족도 향상을 위한 개인화 시스템에 관한 연구, 한국컴퓨터교육학회 논문지, 6권 4호.
- [2] 이용준, 이세훈, 왕창중(2003), 협업 여과 기반의 교육용 콘텐츠 추천 시스템 설계, 한국컴퓨터교육학회 논문지, 6권 2호.
- [3] 강신천, 한승록, 박정환(2003), 교수-학습 지원을 위한 웹 에이전트(web agent)의 개발, 한국컴퓨터교육학회 논문지, 6권 1호.
- [4] A. Zimmermann, M. Specht, and A. Lorenz(2005), Personalization and Context Management, User Modeling and User-Adapted Interaction, Vol. 15, No 3-4, Springer Netherlands.
- [5] 박성준, 김주연, 김영국(2002), 분산 이기종 인터넷 쇼핑몰 환경에서의 벡터 모델 기반 개인화 서비스 시스템, 정보과학회논문지: 컴퓨팅의 실제, 8권, 2호.
- [6] W. Chiu(2001), Web Site Personalization, IBM Technical Library, [Online] available: <http://www-128.ibm.com/developerworks/websphere/library/techarticles/hipods/personalize.html>.
- [7] 김병만, 이경, 김시관, 임은기, 김주연(2004), 추천시스템을 위한 내용기반 필터링과 협력필터링의 새로운 결합 기법, 정보과학회 논문지: 소프트웨어 및 응용, 31권, 3호.
- [8] G. Adomavicius & A. Tuzhilin(2005), Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6.
- [9] Amazon, <http://www.amazon.com/>.
- [10] Movielens, <http://movielens.umn.edu/html/tour/index.html>.
- [11] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl(2003),

MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System, Proceedings of the International Conference on Intelligent User Interfaces.

- [12] J. Basilico & T. Hofmann(2004), Unifying Collaborative and Content-Based Filtering, Proceedings of the 21st International Conference on Machine Learning, Vol.69, ACM Press.
- [13] Y.Z. Wei, L. Moreau, and N.R. Jennings(2005), A Market-Based Approach to Recommender System, ACM Transactions on Information Systems, Vol.23, No. 3.
- [14] 황명권, 공현장, 정관호, 김판구(2005), 온톨로지를 이용한 관광정보 개인화 추천 시스템 설계, 한국컴퓨터종합학술대회 2005 논문집, Vol. 32, No. 1.
- [15] 김범배, 최형기(2005), 신경망과 유전자 알고리즘을 이용한 스팸 메일 필터링 기법의 구현과 성능평가, 정보처리학회논문지, 13-C권, 2호.
- [16] 김동휘, 한이식, 김순자(2001), 에이전트 기반 비교쇼핑 시스템의 개인화 방안, 정보과학회논문지: 소프트웨어 및 응용, 28권, 5호.
- [17] 김영지, 문현정, 옥수호, 우용태(2002), 사례기반추론 기법을 이용한 개인화된 추천시스템 설계 및 구현, 정보처리학회논문지, 9-D권, 6호.
- [18] 이건호, 이동훈(2004), 사례기반추론과 규칙기반추론을 이용한 e-쇼핑몰의 상품 추천 시스템, 정보처리학회논문지, 11-D권, 5호.
- [19] 김형일(2004), 협동적 필터링을 위한 데이터 블러링 기법, 박사학위논문, 동국대학교 대학원.
- [20] 고수정(2005), 사용자-상품 행렬의 최적화와 협력적 사용자 프로파일을 이용한 그룹의 대표 선호도 추출, 정보과학회논문지: 소프트웨어 및 응용, 32권, 7호.
- [21] T. Husted, C. Dumoulin, G. Franciscus, and D. Winterfeldt(2002), Structs 자바 프레임워크를 이용한 웹 애플리케이션의 구축(Structs In Action), 남기혁, 지상준 공역(2004), 인포북.
- [22] M.J. Pazzani(1999), A Framework for Collaborative, Content-Based and

Demographic Filtering, Artificial Intelligence Review, Vol.13, No.5-6,
Springer Netherlands.

- [23] 김형일, 김준태(2005), 협동적 여과에서의 희소성 문제 해결을 위한 데이터 블러링 기법, 정보과학회논문지: 소프트웨어 및 응용, 32권, 6호.
- [24] 이동국(2002), CRM 프로세스를 적용한 개인화 된 여행안내 전문가시스템, OA 학회 논문지, 7권, 1호.
- [25] 성경상, 박연출, 안재명, 오해석(2004), 고객의 선호 특성 정보를 이용한 상품 추천 시스템, 정보처리학회논문지, 11-D권, 5호.
- [26] J. Breese, D. Heckerman, and C. Kadie(1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Technical Report MSR-TR-98-12, Microsoft Corporation
- [27] E-Y. Kang, H. Kim, J. Cho(2006), Personalization Method for Tourist Point of Interest (POI) Recommendation, Lecture Notes in Artificial Intelligence, Vol.4251.

<Abstract>

**Development and Implementation of a Framework
for Personalized Recommender System based on Hybrid Filtering**

Kang, Eui-Young

Computer Education Major
Graduate School of Education, Cheju National University

Supervised by Professor Cho, JungWon

Personalization breaks down the abundance of information in today's society and effectively allows users to obtain information that is corresponded to their needs, goals, knowledge, interests or other characteristics. Thus, personalized service reduces lost time and effort and increases satisfaction. As it creates a positive effect, personalized service can be applied to various practices. In education, the target is students and provides for learner-centered curriculum. This encourages a learner to become active in the learning process and become a better scholar. In business, it allows companies to personalize their targeting of customers and increasing customer loyalty, as reflected in the revisiting rates and program reusability. However, in personalized services, the provider puts in a lot of time and effort to classify a large amount of data and build up the system. This process must be repeated often as the focus of the personalized service changes and must be adapted into the system.

Therefore, this thesis proposes a framework for personalized service that can be reused regardless of the targeted item. The proposed framework provides personalized information according to user preferences with structured attributes for each item. To achieve this solution, this thesis analyzes the existing system concerning personalization as well as the framework design and suggests the

application of key technologies. In applying this framework, there are two suggested applications: one detailing a personalized subject recommender system for lectures and another presenting a personalized tourist recommender system. Each of these systems is evaluated performances using hit-ratio statistics. Overall, the proposed framework shows a good performance for these personalized service applications.

