연관마이닝에 의한 데이터베이스캐시 설계
(Design of Database Cache by Association Mining Method)

사 재학*, 남 인길**
(Jae-Hak Sa In-Gil Nam)

요 약 효율적인 데이타마트 정보의 축척과 질의 정보 추출을 위한 연관 마이닝 방법을 적용하여 검색 속도를 빠르게 할 수 있도록 테이블을 생성하고 고객의 속성별 가중치와 선호기준을 입력받아 선호 점수를 계산하여 점수가 높은 과목을 우선적으로 검색할 수 있도록 기존 연관 알고리즘에서 사용한 단일 항목 입력 데이터 구조를 확장하여 다중 항목 연관 알고리즘(Multiple Item Association Mining : MIAM)을 이용하여 생성된 연관 검색 유형 테이블을 데이터베이스캐시화를 설계하였다. 동일한 알고리즘에서도 데이터베이스캐시 시스템을 적용한 시스템의 질의 처리 수행속도가 우수성을 이용하여 설계함으로써 효율적인 웹 서버 기능을 수행할 수 있음과 동시에 데이터베이스 캐싱의 주요 이점인 효율성 증대, 속도 향상, 비용절감의 효과를 얻을 수 있으므로 연구 설계하였다.

1. 서 론

1.1 연구 배경

현재 인터넷 쇼핑몰에서 제공하는 서비스의 개념은 예전의 대량생산에 따른 시장점유의 개념에서 차별화 된 고객점유의 개념으로 바뀌고 있는 추세이며 이를 위해서는 고객의 정적인 정보와 더불어 고객의 행동을 시스템이 기억하고 이를 바탕으로 고객의 다음 행동을 예측하거나 관련성 있는 고객들 사이에서 나타나는 연관성을 바탕으로 고객에게 차별화 된 서비스를 제공하는 기능이 필요하다. 이것은 가상 공간에서 고객과의 만남을 넓히고 구매로 이끌기 위해서는 고객의 기호, 구매 패턴, 경제력 등의 데이터가 다각도로 분석되어야 하기 때문이다.

이러한 현실적인 정보를 제공할 수 있는 데이터 마이닝에 의한 데이터 마트의 성장과 연관 알고리즘(Algorithm)에 의해 정해진 과거 데이터로부터 입력과 결과사이의 어떤 패턴 관계가 있는가를 찾아내고 이를 바탕으로 미래의 결과를 예상함으로써 보다 효율적인 의사결정을 지원하고 정보의 축척과 양질의 정보를 효율적으로 제공하기 위한 방안으로 연관 마이닝에 의한 데이터베이스캐시 시스템을 연구하였다.

*대구대 대학원 컴퓨터정보공학과 전산공학전공
**대구대학교 사무처장

1.2 연구 목적 및 방법

데이터 마이닝은 대량의 데이터에서 알려지지 않은 데이터 규칙 및 패턴을 발견하는 데이터 분석 기술이다[5,8,9]. 데이터 마이닝의 방법에는 연관, 의사결정 트리, 신경망 이론 등이 있다. 이 중에서 연관 방법은 데이터 집합을 조사하여 데이터간의 연관성이 있는 집합을 발견하는 것이다. 이 방법은 초기 연구에서 상품간의 연관성을 발견하여 아이템 배치와 판매촉진 전략 등 장바구니 분석에서 가장 널리 사용되고 있다[4].

AIS, SETM, Apriori등의 기존 연관 마이닝 알고리즘은 단일 항목(Transaction_id, Item)을 가진 입력 데이터 구조에서 수행된다[3,44,6]. 이러한 방법들이 상품 검색에서 이용될 경우에 레코드가 많이 발생되어 연관 집합을 발견하는 데 수행 속도가 떨어지는 문제점이 생긴다.

이러한 단점을 보완하기 위해 황현숙은 상품 선호 기반의 의사 결정 모델을 개발하고 다속성 의사결정 모델 중에서 점수 모델을 사용하여 기존 연관의 마이닝 알고리즘은 단일 항목을 가진 입력 데이터 구조에서만 수행되는 것을 확장하여 다중 항목 필드 데이터 구조에서 연관 집합을 생성한 후 다중항목 알고리즘을 개발 적용하여 단일 항목 검색보다 다중 항목 검색에서 검색 속도의 향상된 연구 결과를 가져왔다[1].

본 논문에서는 선행 연구에 의해 생성된 물리적 정보 저장소를 논리적 환경의 메모리 영역에

상주시킴으로서 보다 향상된 정보 검색을 지원하고 시스템의 효율적 이용과 성능향상 효과를 가져올 수 있다고 판단하여 가상대학의 수강신청시스템을 대상으로 연구 설계하였다.

2. 선행 연구

2.1 데이터 마이닝

데이터 마이닝에 대한 정의를 가트너 그룹(Gatner group)에서는 "기업이 보유하고 있는 대용량의 데이터에서 통계적 방법이나 모델링 기법과 같은 패턴인식기술을 이용하여 기존에 발견되지 않고 숨겨져 있던 의미있는 데이터 사이의 관계"라 정의하였으며, Aaron Zornes, The META Group에서는 "데이터 마이닝은 매우 큰 데이터베이스로부터 사전에 알려지지 않은, 유용한 정보를 추출하는 지식 발견 방법이다" 라고 정의하였다.

2.2 연관 알고리즘

2.2.1 AIS 알고리즘

Agrawal et al.은 장바구니 데이터를 대상으로 고객이 구매한 상품간에 연관성이 있는 집합을 발견하는 AIS(Association Item Sets) 알고리즘을 제안하였다[2]. AIS 알고리즘은 전체 데이터베이스를 검색하여 최소한의 트랜잭션 개수를 후보 항목 집합을 발견하였다. 또한 연관 규칙 생성에 대한 기준으로 사용하고 있는 최소 지지도와 신뢰도 설정에 대해서 논의하였다.

Houstma and Swani의 SETM(Sets Mining) 알고리즘은 AIS 알고리즘을 기반으로 데이터베이스 질의문을 사용하여 연관 집합을 발견하였다[6]. 그러나 AIS와 SETM 알고리즘은 후보 집합을 생성할 때 데이터베이스를 여러 번 접근하여 생성하기 때문에 메모리 관리와 성능에서 효율적이지 못하였다.

2.2.2 Apriori 알고리즘

Agrawal and Srikant는 이전 단계의 빈발 함수 집합을 이용하여 후보 집합을 생성하는 Apriori 알고리즘을 제안하였으며[3], AIS와 SETM 알고리즘에 비해 실행 시간에서 더 좋게 나타났다. 수행속도가 빠르다고 검증된 Apriori 알고리즘의 연관 규칙 생성은 크게 두 단계로 나눌 수 있다. 첫 번째 단계에서는 최소 지지도 이상을 갖는 빈발 항목 집합(large item sets)을 발견한다. 두 번째 단계에서는 발견된 빈발 항목 집합의 모든 부분집합을 생성하여 최소 신뢰도 이상인 규칙을 발견한다.

Apriori 알고리즘은 지지도를 효율적으로 계산하기 위해 해쉬-트리(hash-tree) 데이터 구조를 사용하여 Ck를 저장하고 지지도를 계산하기 위해서는 자신만의 데이터 구조를 필요로 하고 제거 과정과 트랜잭션 개수를 계산할 때 질의문을 사용하지 않는 단점을 가진다.

성능이 우수하다고 평가받은 Apriori 알고리즘은 장바구니 분석, 침입 시나리오 자동 생성, 의료정보시스템 등에서 응용되었다[5,11,12,21].

2.2.3 단일 항목 필드 구조의 SQL 기반 연관 알고리즘

Sarawagi et al.은 후보 집합 생성과 트랜잭션 개수를 계산하기 위해 Apriori 알고리즘을 기반으로 데이터베이스 시스템과 통합한 알고리즘을 제시하였다[10]. Apriori 알고리즘의 성능을 개선하려는 여러 알고리즘[13]이 연구되고 데이터 흐름의 기본 구조는 유사하였다.

Sarawagi가 제시한 알고리즘을 살펴보면 입력 데이터 구조는 트랜잭션을 구분하는 트랜잭션 번호(Tid)와 데이터 항목을 나타내는 한 개의 항목필드로 구성된 단일 항목 필드 구조(Tid, Item)로 구성되어 있다.

Agrawal의 초기 AIS 알고리즘 이후 다수의 연구자들은 규칙을 생성하기 위해 후보 항목 집합을 효과적으로 구하는 알고리즘에 대해 연구하였고[3,14,15], 그 외 다수의 응용 알고리즘을 제안하였다[5,13,15]. 그리고 데이터 마이닝의 질의문에 필요한 새로운 연산자를 제안하였다[16,17,18]. 특히 Apriori 알고리즘은 수행 성능이 좋은 것으로 평가를 받아 여러 분야에서 기본 알고리즘으로 활용되었다.

<표 1> 연관 알고리즘 비교

알고리즘	특 징
AIS(Agrawal et al., 1993)	초기의 연관 알고리즘으로 연관성이 있는 집합을 발견
SETM(Houtsma and Swani, 1995)	쿼리문으로 연관 집합을 발견하였으나 성능에서 효율적이지 않았음
Apriori(Agrawal and Srikant, 1994)	이전 단계의 후보 항목 집합에서 빈발 항목 집합을 생성
SQL 기반의 알고리즘(Sarawagi et al., 2000)	단일 항목 필드(T_id, Item)를 가진 입력 데이터 구조에 SQL 기반의 알고리즘 제시

<표 1>은 앞에서 분석한 기존 연관 알고리즘인 AIS, SETM, Apriori, SQL 기반의 알고리즘의 특징을 비교한 것이다.

2.2.4 다속성 의사결정 모델

조직의 경영에서 발생될 수 있는 다양한 문제에 대해 최적의 해결책을 얻기 위해 경영과학 모델을 사용하게 된다. Blanning, BanerJee and Basu는 경영과학 모델은 문제의 유형에 따라 적용하는 방법이 여러 가지가 있으며, 문제의 유형에 따라 예측 모델, 최적화 모델, 대기 행렬 모델, 재고 모델, 회계 모델, 재무 모델, 다기준 의사결정 모델 등으로 분류하였다[19,20].

2.2.5 다중 항목 알고리즘

기존의 연관 알고리즘이 단일 항목 입력 데이터 구조에서 연관 집합을 생성하는 문제점을 보완하기 위해 다중 항목의 입력 데이터 구조에서 적용할 수 있는 연관 마이닝 알고리즘이다[1].

1)다중 항목 필드 구조의 연관 알고리즘

(1) 다중 항목 필드 구조

단일 항목 필드 구조의 연관 알고리즘은 장바구니 분석에서 주로 응용되었다[5,21]. 수강신청 분석은 학생이 신청한 과목 데이터를 대상으로 연관성을 가지는 과목을 발견하는 것이다. <표 2>은 수강신청 분석에서 사용한 입력 데이터 구조이다. 트랜잭션 번호(T_id)는 학생별로 신청한 과목에 대해 동일한 번호를 부여하고 있으며 항목(Item)은 신청한 과목을 나타낸다. 이러한 입력 구조에서는 학생이 가상대학에서 여러 가지 과목을 신청할 경우, 학생이 신청한 과목의 개수만큼 레코드가 생긴다.

다중 항목 구조는 두 가지 유형이 존재할 수 있다. 첫 번째 경우는 동일 항목 필드에서 동일한 속성이 없을 경우로 수강신청 분석 데이터가 이에 속한다. <표 3>은 수강신청 데이터에서 단일 항목 구조를 다중 항목(ITEM1, ..., ITEMn) 구조로 변경하는 것을 나타내고 있다. 다중 항목의 데이터는 단일 항목 구조의 동일한 트랜잭션 번호에 대해 한 개의 레코드가 생성된다. 이 경우 다중 항목 필드 구조로 변경할 때 항목 데이터의 입력 순서는 데이터의 카운트를 구하기 위해서 레코드별로 일관성을 가져야 한다. 예를 들어 <표 3>에서 단일 항목 필드의 트랜잭션 번호가 1일때는 OS, DS, DB의 순이고 3일때는 DS, OS의 순으로 데이터를 저장하고 있다. 이를 다중 항목 데이터 구조로 변경할 때 항목 순서가 일관성을 가지기 위해서 다중 필드의 3번 트랜잭션은 OS, DS 순으로 저장되어야 한다.

<표 2> 수강신청 분석 입력 데이터 구조

T_id	item
1	OS
1	DS
1	DB
2	DS
2	DB
2	SE
3	DS
3	OS

<표 3> 수강신청 분석의 다중 항목 필드 구조

단일 항목 필드

T_id	Item
1	OS
1	DS
1	DB
2	DS
2	DB
2	SE
3	DS
3	OS

다중 항목 필드

T_id	$Item_1$	$Item_2$	$Item_3$
1	OS	DS	DB
2	DS	DB	SE
3	OS	DS	

```
Algorithm Multi_Item_Association
n : 전체 항목 개수; k : 각 단계의 색인
trans_tbl : 트랜잭션 테이블; tr_count : 총 트랜잭션의 개수
c.count : 각 후보 항목 집합의 트랜잭션 개수
ikind : 동일 필드의 속성(1:동일한 속성)
for (k=1;  n;  k++ ) do begin
   Ck = Candidate_Item_Generation(trans_tbl, k, n, ikind);    //k개 후보 항목 집합 생성
   Fk = Frequent_Item_generation(Ck, k, n);                   // k개의 빈발 항목 집합 생성
end
for (k=2; Fk<0; k++) do begin
   Rk = Rule_Set_Generation(Fk, tr_count);                    // 지지도를 상속 받은 신뢰도 규칙 집합 생성
end
```

<그림 1> 다중 항목 연관 알고리즘

둘째는 동일 항목 필드에서 동일한 속성이 있을 경우로 이는 과목 검색 유형 분석 데이터가 이에 속한다. 과목 검색 유형 데이터는 가상대학에서 학생이 키워드로 검색한 과목의 속성에 관한 데이터를 말한다.

(3) 다중 항목 연관 알고리즘

본 논문에서는 앞에서 가정한 두 가지 유형에 대해 다중 항목 필드 구조에서 연관 규칙을 생성하는 다중 항목 연관 마이닝(Multiple Item Association Mining : MIAM)알고리즘을 다음과 같이 구성하였다. 다중 항목 연관 마이닝의 메인 알고리즘은 <그림 1>과 같이 3개의 함수를 가진다. 첫 번째는 후보 집합을 생성하는 Candidate_Item_Generation()함수로서 이의 결과는 Ck테이블에 저장된다. 두 번째는 후보 항목 집합인 Ck에 대해 최소 지지도 이상을 가지는 빈발 항목 집합을 생성하는 Frequent_Item_Generation()함수로서 이의 결과는 Fk테이블에 저장된다. 마지막으로 빈발 항목 집합인 Fk에 대해 각 단계에서 최소 신뢰도 이상을 가지는 연관 규칙 집합을 생성하는 Rule_Set_Generation()함수로서 이의 결과는 Rk에 저장된다.

2) 후보 항목 생성 알고리즘

입력 테이터 테이블인 트랜잭션 테이블(Trans_tbl)은 n개의 항목 필드를 가지고 잇다. 여기서 k 단계의 후보 항목 집합은 트랜잭션 테이블의 n개의 항목 집합에서 k개의 모든 항목 집합으로 구성한다. 예를 들어 k=1일 때 후보 항목 집합은 트랜잭션 테이블의 n개의 항목에서 1개의 항목으로 구성할 수 있는 모든 집합이고 k=2일 때 n개의 항목에서 2개의 항목으로 구성할 수 있는 모든 집합을 의미한다. k=n일 때 후보 항목 집합은 트랜잭션 테이블에 있는 n개 항목 모두로 구성할 수 있는 집합을 의미한다.

다중 항목 연관 마이닝 알고리즘에서 k단계의 후보 항목 집합은 다음과 같은 과정으로 생성한다. k 단계의 후보 항목 집합은 트랜잭션 테이블에서 나오는 k개의 유형에 대한 질의문에서 생성된다. 예를 들어 다음은 전체 문항수(n)가 4일 경우 단계 k에 따라 생성되는 항목 필드의 구성을 나타낸다.

<그림 2>는 k단계의 후보 항목 집합을 생성하는 알고리즘이다. 트랜잭션 테이블의 n개 항목 집합에서 생성되는 k개의 항목 필드에 해당하는 테이타가 후보 항목 집합이 된다. 이러한 k개의 항목 집합은 질의문에서 그룹화 연산(group by)으로 생성된다.

다중 항목 입력 데이터의 속성은 동일 항목 필드에서 동일한 데이터 속성이 잇는 경우와 없는 경우로 분류하였다. 동일 필드에서 동일한 속성이 있는 경우 (ikind=1)는 k개의 항목 집합에 대해 한번의 그룹화로써 count(*) 함수를 이용하여 레코드의 개수를 구한다. 그렇지 않을 경우는 여러 필드에 분산되어 있는 데이터의 카운트를 구하기 위해 두 번의 그룹화를 수행한다. 두 번째 수행하는 그룹화 연산에서 카운트는 첫 번째 그룹화에서 생성한 k개의 집합(TCk)에서 동일한 집합에 대해 sum(count) 함수의 누적으로 계산된다.

```
Algorithm Candidate_Item_Generation(Treans_tbl, k, n, ikind)
if (ikind=1) then C=Ck
else C=TCk
endif
for (i1=1; n-(k-1); i1++) do bigin
  for(i2=i1+1; n-(k-2); i2++) do begin
    for(i3=i2+1; n-(k-3); i3++) do begin
                    .....
      for(ik=ik-1+1; n; ik++) do begin
        insert into C
          select item1, Item2, Item3, ..., Itemk, count(*) from Trans_tbl
                group by Item1, Item2, ..., Itemk      order by Item1, Item2, ..., Itemk
      end
            - - -
    end
  end
end
if (ikind≠1)then
  insert into Ck
      select Item1, Item2, ... , Itemk, sum(count) from TCk
            group by Item1, Item2, ..., Itemk      order by Item1, Item2, ..., Itemk
endif
```

<그림 2> 후보 항목 집합 생성 알고리즘

3) 빈발 항목 생성 알고리즘

k 단계의 빈발 항목 집합은 k 단계의 후보 항목 집합 중에서 최소 지지도 이상의 카운트를 가지고 이전 단계의 빈발 항목 집합인 Fk-1의 부분 집합인 집합으로 구성된다. 이를 위한 질의문은 후보 항목 집합인 Ck와 Fk-1과 k번

```
Algorithm Frequent_Item_generation(Ck, k, n)
if (k=1) then
  insert into Fk
    select item1, item2, ..., itemk, cnt, from Fk  where cnt ≥ Mn*k(support)
else
 insert into Fk
    select C.item1, C.item2, ..., C.itemk, C.cnt   from Ck as C, Fk-1 I1, ..., Fk-1 Ik
    where C.item1=I1.item1 and C.item2=I2.item2 and ... and  C.itemk-1=Ik.itemk-1 and
        C.item1=I1.item1 and C.item2=I2.item2 and ...  and  C.itemk=Ik.itemk-1 and
                              .....
        C.item1=I1.item1 and C.item2=I1.item2 and ... and  C.itemk-1=Ik.itemk-2 and C.itemk=Ik.itemk-1 and
        C.item1=I1.item1 and C.item2=I1.item2 and ... and  C.itemk=Ik.itemk-1 and
        C.item2=I2.item1 and C.item3=I1.item2 and ... and  C.itemk=Ik.itemk-1 and  C.cnt ≥ Mn*k(support)
endif
```

<그림 3> 빈발 항목 집합 생성 알고리즘

조인을 수행하여 k-1 단계의 빈발 항목 집합에 속하는 부분을 추출하는 조건을 포함하여야 한다. 질의문의 조건에서는 k개에서 k-1개의 항목의 유형에 대해 Fk-1의 항목과 같은지를 비교한다. 즉, k개 유형의 각 후보 항목 집합을 Fk-1의 k-1개 유형과 같은지를 비교한다. 예를 들어 k=3일 때 Ck에서 Fk-1에서 부분집합에 속하는 후보 집합을 추출하는 질의문은 다음과 같다.

<그림 3> 은 빈발 항목 집합을 생성하는 알고리즘이다.

```
select item1, item2, item1, item1 from C3 C, F2 I1, F2 I2, F2 I3
    where C.item1=I1.item1 and C.item2=I2.item2 and  C.item1=I1.item1 and C.item3=I3.item2 and
        C.item2=I2.item1 and C.item3=I3.item2
```

k가 1일 때 빈발 항목 집합은 항목 집합의 카운트가 최소 지지도 이상인 조건을 만족하는 질의문에서 생성된다. k가 2이상일 때 빈발 항목 집합은 이전 단계의 빈발 항목 집합인 Fk-1의 부분 집합을 추출하는 조건과 항목 집합의 카운트가 최소 지지도 이상인 조건을 만족하는 질의문에서 생성된다.

4) 연관 규칙 생성 알고리즘

최소 신뢰도 이상을 가지는 신뢰도 규칙 집합을 생성하기 위해서는 다음 2개의 과정이 필요하다.

첫째, k 단계의 빈발 항목 집합에 대해 규칙의 조건부와 결과의 항목 순서를 저장하는 규칙 집합 테이블이 필요하다.

둘째, 규칙 집합의 신뢰도를 계산하여 최소 신뢰도 이상인 집합을 연관 규칙 집합으로 한다.

<표 4>은 각 단계의 빈발 항목 집합에서 생성되는 모든 가능한 규칙을 나타낸다. 여기에서 집합의 유형은 규칙의 조건부와 결과에서 구성할 수 있는 항목의 수를 의미하며 이러한 집합 유형은 k 단계에 따라 k-1개의 유형으로 구성된다. 예를 들어집합 유형은 (1, 3), (2, 2), (3, 1) 형태로 구성된다. 규칙의 개수인 kCi는 k개 중에서 조건부인 i개 유형을 추출하는 조합의 개수를 의미한다. 여기서 i는 조건부에 있는 항목 집합의 개수를 의미한다. N(Fk)를 n단계의 빈발 항목 집합의 개수라 두면, 2번째 단계부터 n단계까지의 빈발 항목 집합에서 발생하는 전체 규칙 집합의 개수는 다음과 같이 계산된다.

전체 규칙 집합의 개수 =

<표 4> F_k에서 생성하는 모든 규칙 집합

빈발 항목	집합 유형	생성규칙(item1 a, item2 b, ---)	규칙개수
F_2	(1,1)	a=> b b=> a	$_2C_1$
F_3	(1,2)	a=> b,c b=>a,c c=>a,b	$_3C_1$
	(2,1)	b,c=>a, a,c=>b, a,b=>c	$_3C_2$
F_4	(1,3)	a=>b,c,d b=>a,c,d c=>a,b,d d=>a,b,c	$_4C_1$
	(2,2)	a,b=>c,d a,c=>b,d a,d=>b,c c,d=>a,b b,d=>a,c b,c=>a,d	$_4C_2$
	(3,1)	b,c,d=>a a,c,d=>b a,b,d=>c a,b,c=>d	$_4C_3$
		:::::::::::::::::::::::	
F_n	(1, n-1)	item1=>item2, item3, ..., itemn, ..., itemn=>item1, item2, ..., itemn-1	$_nC_1$
	(2, n-2)	item1, item2=>item3, item4, ..., itemn,	$_nC_2$
	...	.....	...
	(n-1, 1)	item2, item3, ..., itemk=>item1, ...	$_nC_{n-1}$

```
select * from rst3;

item1  item2  item3
-----  -----  -----
item1  item2  item3
item2  item1  item3
item3  item1  item2
item1  item2  item3
item1  item3  item2
item2  item3  item1
(6 row(s) affected)
```

```
select * from rst4;

item1  item2  item3  item4
-----  -----  -----  -----
item1  item2  item3  item4
item2  item1  item3  item4
item3  item1  item2  item4
item4  item1  item2  item3
item1  item2  item3  item4
item1  item3  item2  item4
item1  item4  item2  item3
item2  item3  item1  item4
item2  item4  item1  item3
item3  item4  item1  item2
(10 row(s) affecte)
```

<그림 4> F_3, F_4의 규칙 집합 테이블

$$\sum_{k=2}^{n} N(Fk) * (kC1 + kC2 + - - + kCk - 1)$$

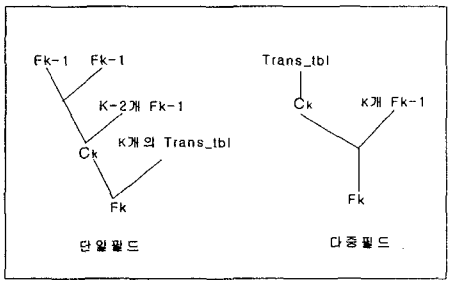
k 단계에서 생성 가능한 모든 규칙 집합을 저장하고 있는 테이블을 RSTk라고 정의하면 <그림 4>은 3단계와 4단계에서 생성되는 모든 규칙의 집합을 저장하고 있는 RST3과 RST4 테이블을 나타낸다. F3의 조건부와 결과의 항목 개수의 유형은 (1,2)과 (2,2) 형태이고 이때 RST3은 전체 6개의 레코드를 가진다. F4의 조건부와 결과의 항목 개수의 유형은 (1,3), (2,2), (3,1)형태이다. 이 유형중에서 (3,1)의 유형은 (1,3)의 역의 형태이므로 RST4 테이블은 (1,3)과 (2,2) 유형에 대한 항목 집합으로 구성하면 된다.

<그림 5>는 규칙 항목의 순서를 가진 RSTk와 빈발 항목 집합 테이블인 Fk를 이용하여 최소 신뢰도 이상을 가지는 규칙 집합 테이블인 Rk를 생성하는 알고리즘이다. 색인 i는 단계 k의 조건부 항목 집합의 개수를 나타내고 j는 조건부 i개 항목에서 생성되는 항목 유형의 개수를 나타내는 색인이다. jj는 레코드 순서를 누적하기 위한 색인으로 사용된다.

규칙 집합인 Rk의 지지도는 빈발 항목 집합 Fk에서 이미 상속을 받고 있기 때문에 Rk는 최소 신뢰도만을 고려하여 생성하면 된다. 신뢰도 규칙 집합은 RSTk 테이블에서 항목의 순서에 대한 값을 추출한 후 Fk 테이블을 이용하여 신뢰도를 계산하여 최소 신뢰도 이상인 집합을 가지게 된다. 알고리즘에서 지지도는 Fk의 카운트를 전체 트랜잭션 개수로 나눈 값이고 신뢰도(confidence)는 Fk의 조건부 항목 개수까지인 Fi의 카운트로 나눈 값이다.

5) 다중항목 필드 구조의 효율성 검토

제시한 다중 항목 필드 구조가 단일 항목 필드 구조보다 질의문의 수행속도가 어느 정도 효율성이 있는지 실험하였다. <그림 6>은 단일 항목 필드와 다중 항목 필드



<그림 6> 연관 집합 생성 과정

```
Algorithm Rule_Set_Generation(Fk, tr_count)
for(i=1; k/2; i++) do begin
 for(j=1; kCi; j++) do begin
  jj=jj+j;
   item1 = RSTi(jj).rs1; item2 = RSTi(jj)rs2; ... ; itemk = RSTK(jj).rsk;
   insert into Rk
    select item1, item2, ..., itemi as "itemi =>", itemi+1, ..., itemk,
      cnt/tr_count as support,  cnt/(select cnt from fi
          where fi.item1=fk.item1, ..., fi.itemi=fk.itemi,) as confidence
          from fk  where confidence ≥ MINk(confidenc)
   if (i≠k/2) then
    insert into Rk
      select itemi+1, ..., itemk as "itemk =>" item1, item2, ..., itemi+1, ...,
       fk-1.itemk-i=fi.itemk,) as confidence
       from fk where confidence ≥ MINk(confidence)
     endif
   end
end
```

<그림 5> 상속된 지지도에 대한 신뢰도 규칙 생성 알고리즘

구조에서 연관집합이 생성되는 과정이다.

단일 항목 필드 구조에서 후보 항목 집합은 이전 단계 Fk-1의 조인으로 생성되고 빈발 항목 집합은 Ck와 k개의 트랜잭션 테이블과의 조인으로 생성된다. 다중 항목 필드 구조에서 후보 항목 집합은 트랜잭션 테이블의 조회로 생성되고 빈발 항목 집합은 Ck와 k개의 Fk-1 테이블의 조인으로 생성된다. 이러한 구조에서 보면 단일 항목 필드 구조에서는 레코드 개수가 많은 트랜잭션 테이블과의 k번 조인으로 실행 시간이 많이 걸리게 된다. 하지만 다중 항목 필드 구조에서는 트랜잭션 테이블의 조회에서 후보 항목을 생성함으로 수행 시간이 단축된다.

3. 데이터베이스캐시의 개념과 구조

3.1. 웹 사이트의 문제점

첫째, 레거시(Legercy) 솔류션 기능과 현재 기술과의 이식성 및 호환성 부족으로 연동이 어렵고 하드웨어적인 특성을 살리지 못한 상태이며, 둘째로는 단순한 정적인 컨텐츠의 제공으로 1세대 웹 사이트에서 오늘날처럼 동적인 멀티미디어 데이터에 대한 기술 포팅 부재로 사용자 요구를 충족시키지 못할 뿐만 아니라 성공적인 온라인 비즈니스 모델에는 동기적으로 생성되는 데이터베이스 중심적인 핵심적인 역할을 기능을 지원하지 못하고 있는 실정이다. 이러한 레거시 시스템의 성능을 향상시키기 위한 대안으로 하드웨어의 자원 증대 필요 요인을 가져다 주고 있기 때문이다.

3.1.1 브라우저 캐싱의 한계

모든 사이트에서 제공되는 모든 컨텐츠들을 전 세계의 PC에 설치된 수백만개의 웹 브라우저의 부라우저 캐시에 저장할 수 만 있다면 전 세계 모든 웹 사이트의 속도는 엄청나게 빨라질 것이지만 이를 위해서는 모든 사용자가 슈퍼컴퓨터를 보유해야 하는 불합리한 일이며 모든 분산 컨텐츠를 관리하는 일도 불가능할 것이다.

3.1.2 프록시 캐싱 자원의 한계

프록시 캐시 서버는 주로 기업 방화벽의 에지에 설치되거나 ISP(Internet Service Provider)에서 호스팅되는 서버 기반 솔류션이다. 이들 서버는 수 많은 브라우저 클라이언트와 공중 인터넷 사이에 설치되고 프록시 서버는 사용자가 웹 사이트에 액세스하고자 할 경우 요청된 객체를 로컬 캐시에서 찾아본 후 이 객체가 존재할 경우 이를 즉시 클라이언트로 전송한다. 프록시 캐싱의 가장 큰 목적은 기업 및 ISP를 위해 대역폭 소모를 감소시키는 역할은 하지만 웹 사이트의 소유주들이 직면하고 있는 성능 문제에 대한 적절한 해결방안은 될 수 없기 때문이다.

3.1.4 컨텐츠 서비스 한계(CDN: Content Delivery Network)

CDN 서비스는 웹 사이트의 컨텐츠를 한 군데 중앙 장소에서 사용자에게 가까운 장소로 이동 배치시키는 방식이다. 네트워크에 의한 가장 인접한 컨텐츠를 캐싱할 수 있다. 이 기술은 원래 네트워크의 로드를 감소시키기 위해 설계됐기 때문에 성능 문제에 대한 해결은 어디까지나 부수적인 수준일 뿐인 것이다.

3.1.5 솔류션의 패키화의 한계

웹 캐싱 접근방법들의 한계 때문에 대부분의 웹 사이트들은 맞춤형 솔류션을 구축하고는 있지만 소프트웨어 벤더들의 다양한 제품들을 인한 상호 호환성 문제점이 노출되고 있기 때문이며 이를 위해 동일 제조사의 제품을 일관된 되게 구입 활용하게 될 경우 소프트웨어 제조사에 대한 종속될 우려가 있다. 또한 제품 대체로 인한 추가 경비 부담의 증가로 기업 경영의 자금 압박요인으로도 대두될 수 있기 때문이다.

3.2 웹 사이트의 개선 방안

3.2.1 웹 캐싱

웹 캐싱과 데이터베이스 캐싱은 상호 보완적인 방식으로 사이트의 성능을 강화한다. 웹 캐싱은 페이지 수준에서 정적 및 동적인 컨텐츠의 처리를 가속화함으로서 백 엔드 웹 서버, 애플리케이션 서버, 데이터베이스의 로드를 완화하며 데이터베이스 캐싱은 중요한 관계형 컨텐츠를 데이터베이스로부터 미들 티어로 이동시킨다.

웹 캐시는 맨 처음 요청되는 웹 페이지의 경우 이 요청을 웹 서버로 전송해 처리 및 포맷하도록 하고 있으며, 이 요청이 반환된 후에야 웹 캐시가 이를 메모리에 저장하게 된다.

3.2.2 데이터베이스 캐싱

데이터베이스 캐싱의 경우 "웹 캐시 비적중"이 보다 빨리 처리되어 백엔드 데이터베이스에 미치는 영향을 감소시키며 이는 자주 액세스 되는 관계형 데이터가 미들티어 데이터베이스 캐시에 상주하기 때문이다. 웹 캐싱과 데이터베이스 캐싱의 결합으로 웹 요청이 보다 신속히 처리되기 때문에 웹 사이트는 많은 비용이 소요되는 인프라 업그레이드를 거치지 않고도 보다 많은 수의 사용자에게 보다 풍부한 컨텐츠를 제공할 수 있게 된다.

3.3 데이터베이스 캐시

웹 사이트가 경쟁에서 우위를 차지하기 위해서는 풍부한 컨텐츠를 개별 사용자의 요구에 맞춰 신속히 제공해야 한다. 이러한 요구를 충족시키는 미들 티어 데이터 캐싱 솔류션으로서 애플리케이션 투명성을 제공하고 확장이 용이하며 중앙 집중화된 운영을 지원해야 할 수 있어야 한다.

3.3.1 데이터베이스 캐시의 역할과 기능

데이터베이스캐시는 다중 티어 환경에서 상용 데이터베이스를 이용하는 웹 사이트 및 여타 애플리케이션의 확장성과 성능을 확장시킨다. 읽기 작업이 빈번히 수행되는 데이터 세트를 미들 티어에서 캐싱함으로서 데이터베이스 질의 성능이 높아진다. 이를 위해 데이터베이스 캐시가 갖추어야 기능으로 투명성, 확장성, 응답성, 용이성, 동기화, 통합성 등이 있다.

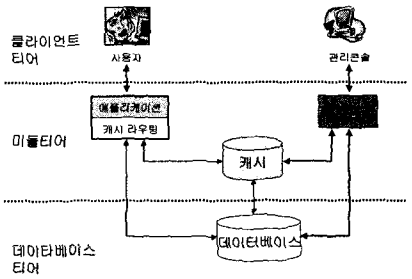
3.3.2 아키텍처

1) 캐시 아키텍처

데이터베이스 캐시는 웹 아키텍처의 미들 티어에 설치된다. 데이터 및 SQL을 애플리케이션과 사용자에 보다 가까운 위치에 처리함로서 성능이 향상되어 네트워크 대기 시간이 짧아진다. 동시에 백엔드 서버의 로드가 완화돼 보다 많은 사용자와 애플리케이션을 지원할 수 있게 된다. <그림 7>에서는 캐시가 읽기 전용 질의를 미들 티어에서 처리함으로서 원래의 데이터베이스의 로드를 감소시켜 보다 많은 사용자를 지원하게 되는 과정을 나타낸 것이다.

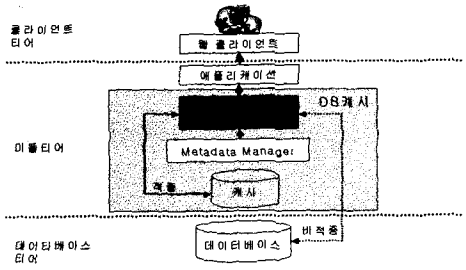
2) 라우팅 아키텍처

데이터베이스 캐시는 JDBC, OLE-DB, ODBC, Pro*C를 이용하는 애플리케이션과 벤더의 Call Interface 라이브러리를 이용해 데이터 액세스 레이어에서 SQL 명령문의 분석 및 라우팅을 처리함으로서 투명성을 지원한다. 미들



<그림 7> Cache 아키텍처

티어 설치 중 라우팅 로직이 포함되며 이 라이브러리는 런타임 중 애플리케이션에 동적으로 연결되며 애플리케이션을 위한 모든 라우팅 기능을 제공한다. <그림 8>는 라우팅 아키텍처를 나타낸 것이다.



<그림 8> 라우팅 아키텍처

3) 관리 아키텍처

관리 아키텍처는 캐시의 통합관리 기능을 제공함으로서 DBA의 기능과 통합되어 있다. 따라서 운영자는 캐싱 작업을 손쉽게 처리할 수 있다.

①캐시로부터 테이블을 추가 혹은 삭제
②캐싱된 테이블을 위한 동기화 정책을 설정
③테이블을 수작업으로 동기화
④성능 및 통계를 모니터

4) 애플리케이션의 구성

캐시는 기존 애플리케이션과 투명하게 연동되도록 설계되었지만 애플리케이션 서버 상에서 운영되는 일부 애플리케이션에 한해 캐시 액세스가 제한되어야 할 경우도 있다.

5) 성능 튜닝

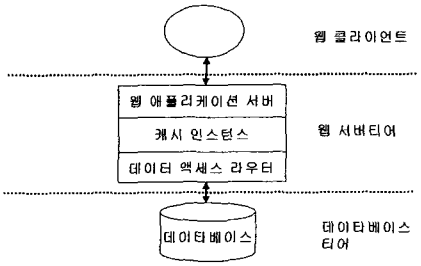
튜닝 프로세스에는 통계 분석, 캐싱할 데이터 세트의 결정, 성능의 평가 등이 포함된다. 원하는 성능을 얻을 때까지 이들 프로세스를 반복해야 할 수도 있다. 다음은 캐시 성능을 향상시키기 위한 방법으로 적절한 애플리케이션 선택, 디스크 액세스의 최소화, 단편화 방지, 테이블 캐싱, 애플리케이션에 의한 연결등을 고려해 볼 수 있다.

6) 관리 콘솔을 통한 성능 모니터링

데이터베이스캐시의 관리기능을 제공한다.

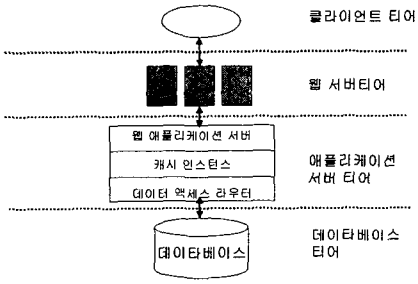
3.4 데이터베이스캐시의 서버 모델 비교

캐시를 설치 구성하는 데는 여러 가지 방식이 있다.



<그림 9> 3티어 모델 구조

일반적으로는 애플리케이션 서버와 데이터베이스가 같은 박스 내에 상주하지 않는 다중 티어 환경에 캐시를 설치한다. 단일 티어 환경에서는 네트워크 대기시간 감소에 따르는 이점이 명확히 드러나지 않을 것이다. 모델 종류로는 <그림 9>의 3티어 모델 구조와 <그림 10>의 4티어 모델 구조가 있다.



<그림 10> 4티어 모델 구조

3.5 데이터베이스 캐시의 설계 목표

미들 티어에서 자주 액세스되는 테이블을 캐싱하므로서 공통 질의에 대한 네트워크 오버헤드를 감소시켜 질의 실행 속도를 단축해주는 성능향상과 확장성, 투명성, 관리 용이성 등 설치와 구성이 단순하고 DBA 관리 기능을 통합 관리 환경을 제공토록 한다.

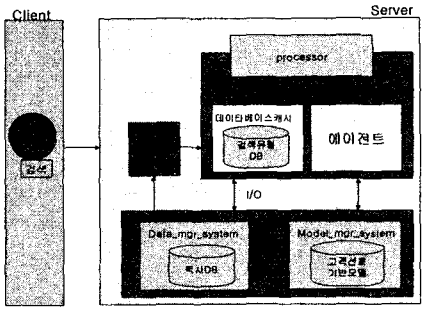
4. 연관 마이닝에 의한 데이터베이스캐시화 설계

본 장에서는 연관 알고리즘 기반위에 학생의 선호 속성을 반영하여 적합한 정보검색의 속도 향상을 위한 데이터베이스캐시를 구현하기 위한 시스템을 설계한다.

4.1 데이터베이스캐시 시스템

4.1.1 데이터베이스캐시 시스템 구조

<그림 11>은 웹 환경에서의 수강신청 과목 검색에서 학생 선호도를 기반으로 데이터마트를 지원 관리하는 데이터베이스 캐시화 시스템의 통합 모듈의 구성을 제시하고 있다. 데이터베이스캐시 시스템의 전체적인 구조의 특징은 첫째, 기 구축된 운영 데이터, 즉 학사운영 데이터베이스를 근간으로 구성되어 있으며 제공자는 학생에게 제공되어질 정보의 속성을 정의하여 데이터베이스캐시 관리기를 이용하여 데이터 마트의 스키마(Schema)정의 및 생성한다. 둘째, 학생의 자주 검색 속성을 서버에서 지원하는 검색 및 모델 에이전트의 도움을 받아 검색의 성향을 분석하여 학생선호 모델 데이터베이스에 저장 관리토록 한다. 셋째, 운영 데이터베이스와 학생의 선호 기반 모델 데이터베이스에서 마이닝한 학생 검색 유형 데이터를 데이터베이스캐시에 적재시키는 구조로 되어 있다.



<그림 11> 데이터마트 개인화 시스템 구성도

본 연구에서 제시한 데이터베이스캐시 시스템 구성요소의 세부적 역할을 기술하면 아래와 같다.

(1) 웹 검색

서비스 요구자와 제공자의 인터페이스 부분으로 GUI 방식에 의한 웹 응용 프로그램으로 작성하여 학생이 원하는 자료의 유형과 검색 키워드를 입력하고 정보를 제공받을 수 있도록 하는 기능을 가지고 있다.

(2) 데이터 관리 시스템(Data Management System)

데이터 관리 시스템은 전체 조직의 운영 데이터를 보관하고 있는 통합 데이터베이스이며, 보다 합리적인 정보검색 서비스를 위해 용도별로 데이터베이스를 구축 운영할 수 있다. 이 운영 데이터베이스는 데이터베이스캐시에서 학생이 원하는 정보를 제공하지 못할 경우, 여기서 검색 정보를 제공한다.

(3) 모델 관리 시스템

모델 관리 시스템은 학생의 선호 검색 정보를 사전 정의하거나 또는 학생이 검색한 정보를 속성별로 가중치와 선호기준을 입력받아 선호 점수를 계산하여 높은 점수를 가진 과목을 우선적으로 제공할 수 있도록 정보의 검색 선호도를 관리하기 위한 자료를 관리 제공한다.

(4) 검색 및 에이전트

검색 및 모델 에이전트는 학생이 인터넷상에서 검색을 수행할 때 적합한 과목을 제공받을 수 있도록 도와주는 역할을 한다.

(5) 캐시 관리기

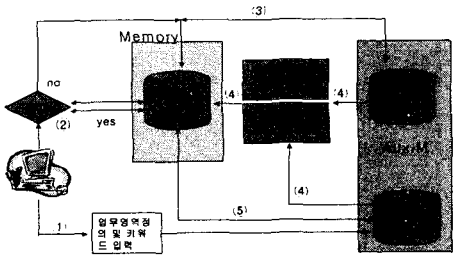
운영 데이터베이스를 기반으로 필요한 테이블과 속성을 선택 또는 신설 속성을 만들어 이를 테이블화하여 데이터베이스캐시에 정의하고 생성하는 관리 기능을 한다.

(6) 데이터베이스캐시

캐시 관리기에 의해 생성된 데이터베이스캐시 테이블로써 학생 선호 기반 데이터베이스와 모델 데이터베이스의 검색 유형을 연관 마이닝에 의해 검색정보를 사전 추출하여 이를 데이터베이스캐시에 적재하여 검색 정보를 서비스한다.

4.1.2 시스템의 모듈 구성

데이터베이스캐시 시스템은 학생이 자주 검색하는 유형을 주기적으로 분석하여 빠른 과목 검색을 제공하기 위한 검색 유형 데이터베이스를 캐시화하고 이를 기반하여 학생이 선호하는 속성을 가지고 상호작용으로 과목검색에 대한 검색 키워드를 지원한다. 이러한 기능을 수행하기 위한 데이터베이스캐시 시스템의 세부 모듈 구성은 <그림 12>과 같다.



<그림 12> 시스템 내부 모듈 처리 과정도

① 업무 속성 및 키워드를 정의 입력한다.

② 필요한 정보 요구와 이에 부합되는 학생 선호 기반 모델에서 속성을 참고하여 검색 결 과를 되돌려 준다. 만약 검색 유형 데이터베이스 캐시 테이블에 필요한 정보가 없을 경우 ③번 항목을 수행한다.

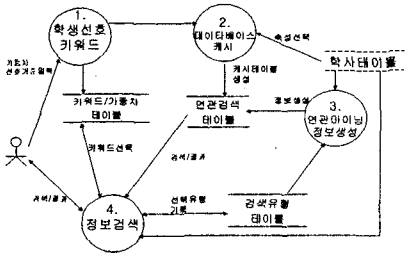
③ 검색 유형 데이터베이스캐시에 원하는 정보가 없을 경우 운영 데이터베이스에서 검색하여 결과를 제공한다.

④ 학생 선호 기반 데이터 모델의 정보와 운영 데이터베이스 정보를 연관규칙 집합생성 모듈에 의해 생성된 데이터베이스캐시 테이블에 적재한다.

⑤ 학생 검색 유형 트랜잭션을 학생 선호 기반 모델에 저장 관리한다.

4.1.3 시스템의 자료 흐름도

<그림 13>은 데이터베이스캐시 시스템의 모듈별 처리 업무에 따른 데이터 입력과 출력, 정보의 흐름, 정보의 저장소를 나타내는 자료 흐름도(DFD:Data Flow Diagram)이다. 직사각형은 시스템 주관 처리 조직을 의미하고 평행

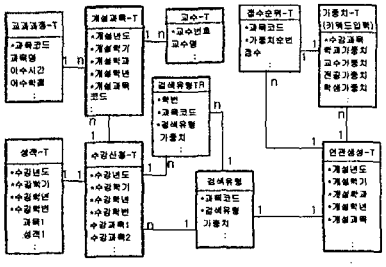


<그림 13> 시스템 자료 흐름도(수준1)

선은 정보의 저장소를 나타낸다. 화살표는 자료의 흐름을 나타내며 원으로 표시한 것은 모듈별 처리 프로세스를 표시한 것이다.

4.1.4 개체 관계도

개체 관계도(Entity Relational Diagram)를 정의하여 세부 모듈에서 테이블을 작성할 때 사용한다. <그림 14>



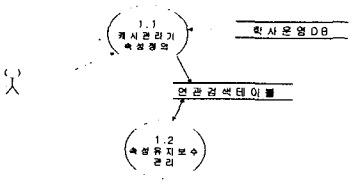
<그림 14> 데이터베이스 개체 관계도

는 키워드 검색 모듈과 연관 마이닝을 이용한 데이터베이스 생성 모듈에서 필요한 개체 사이의 관계도이다.

4.2 시스템 모듈별 상세 설계

4.2.1 데이터베이스캐시 모듈

연관 마이닝에 의해 생성되는 정보의 저장소로 캐시 관리기를 이용하여 물리적 운영 데이터베이스로부터 연관 검색 데이터를 저장할 캐시 테이블의 속성을 선택 정의하여 논리적인 데이터베이스캐시 테이블을 생성한다. <그림 15>은 연관 검색 테이블 생성 자료 흐름도를 나타낸 것이다.



<그림 15> 연관 검색데이타 테이블 생성 자료흐름도(수준2)

(1)데이터베이스캐시 구조

<표 5>는 데이터베이스캐시에 적재된 개설 교과목 테이블을 나타낸다.

<표 5> 개설과목 데이터 테이블

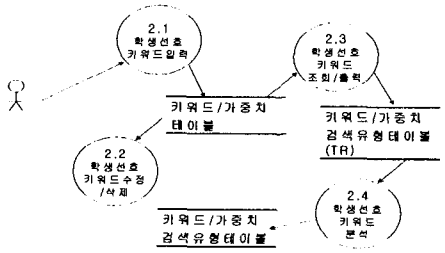
O_year	O_term	O_dept	O_class	O_code	O_prof	O_day	---	O_time	O_wrdat
개설년도	개설학기	개설학과	개설학년	과목코드	강의교수	강의요일	---	수업시수	작성일자

4.2.2 학생 선호 키워드 정의 모듈

<그림 16>는 키워드와 가중치의 속성을 정의 입력하고 이를 유지 관리하여 과목의 속성별 검색이 가능하도록 검색 자료에 대한 자료 흐름도이다.

(1)키워드 정의 테이블 구조

<표 6>은 학생의 검색 선호를 위한 키워드에 대한 가중치를 정의하여 입력 관리하는 테이블이다.



<그림 16> 키워드/가중치 모듈 자료 흐름도(수준2)

<표 7>는 학생들이 선호하여 검색한 키워드 검색 유형을 트랜잭션별로 관리하는 테이블 구조이다. 학생은 수강신청에서 단일 과목에 대해 다수의 검색을 수행할 수 있으며, 학번 트랜잭션은 학생이 수강신청에 접속해서 키워드 과목 검색을 수행할 때마다 누적으로 부여된다.

<표 8>은 <표 7>의 검색 유형 트랜잭션 단위별로 분석하여 검색 유형별 자료를 생성 관리하는 테이블이다. 검색 유형 패턴(Serch_no)은 트랜잭션별 연관 집합을 발견하기 위해서 학생이 검색한 유형을 저장하는 필드이다. 검색 유형 필드에 학생이 검색한 유형을 입력하기 위해서는 모든 속성에서 가능한 검색 유형에 대한 색인이 필요하다.

(2)학생별 검색 유형 데이터

<표 7>는 속성간 검색 유형 트랜잭션 데이터는 수강신청에서 학생이 키워드를 이용하여 과목을 검색할 때 입력한 키워드 데이터이다. <표 8>은 검색 유형 데이터는 학생들이 키워드 검색을 수행한 전체 트랜잭션을 대상으로 속성간의 연관 규칙을 발견하기 위해 필요하다. 속성간 연관 규칙은 최소한 두 개 이상의 속성에서 연관 관계가 이루어진다. 이는 학생이 단일 속성만을 선택할 경우에는 연관 규칙을 반영하지 못하는 단점이 발생한다. 그래서 다음에 제시하는 트랜잭션별 검색 유형 데이터를 사용하여 연관 규칙을 적용함으로써 학생이 단일 항목을 선택할 때도 연관 규칙이 반영될 수 있도록 한다. 즉 n개의 항목에서 생성될 수 있는 모든 검색 유형을 저장하기 위한 테이

<표 6> 키워드 정의 및 가중치 선호 기준 테이블

W_no	W_code	W_prof_w	W_dept_w	W_spec_w	W_stunt_w	—	W_majer	W_maj_cd	W_inde_opt
순번	과목코드	교수 가중치	학과 가중치	전공 가중치	학생선호 가중치	—	전공필수/선택	전공관계코드	산업구분

<표 7> 학생별 검색 유형(TR) 테이블

S_no	A_code	Item1	item2	item3	item4	---	itemn	Serch_no
학번	과목코드	교수 가중치	학과 가중치	전공 가중치	학생선호 가중치	---	전공필수/선택	검색유형

<표 8> 검색 유형 테이블

A_seq	Serch_no	$Item_1$	$item_2$	$item_3$	$item_4$	---	$item_n$
순번	검색유형	교수 가중치	학과 가중치	전공 가중치	학생선호 가중치	---	전공필수 /선택

<표 9> 검색 유형 구분 예제 테이블

S_no	$item_1$	$item_2$	---	$item_n$	S_kind	비고
순번	교수가중치	학생선호 가중치	---	전공필수	검색유형	
①	●		---		S_k(1)	①
②		●	---		S_k(2)	②
③	●	●	---	●	S_k(3)	③

<표 10> 트랜잭션별 검색 유형 테이블

A_code	S_kind1	S_kind2	---	S_kindn
과목코드	첫번째 트랜잭션의 검색유형	두번째 트랜잭션의 검색유형	---	n번째 트랜잭션의 검색유형

블로 <표 9>와 같이 나타난다.

① 교수 가중치 과목을 선택 가능

② 학생 선호 가중치 과목을 선택 가능

<표 9>에서 비고의 ③번 항목은 학생이 수강과목의 모든 항목 가중치를 선택 검색한 유형의 과목이 선정되는 경우를 나타낸다.

이와 같은 검색 유형의 개수는 단일 과목의 경우 <식 1>과 같이 나타난다.

$$n(S\_kind) = \prod_{i=1}^{n} (Itemi(n(Attri\_rw)+1)) - 1 \ldots\ldots \langle \text{수식 1} \rangle$$

<식 1>에서 n(S_kind)는 전체 검색 유형의 개수이고, i는 단일 과목에서 선정할 수 있는 속성별로 선택할 수 있는 키워드의 개수를 나타낸다. 그리고 "n(Attri_kw)+1"은 각 항목에서 선정할 수 있는 속성의 키워드 수를 나타내는 데 여기에 1를 더한 이유는 속성이 선택되지 않은 경우를 포함해야 하기 때문이다. 따라서 전체 검색 유형의 개수는 각 항목에서 가질 수 있는 유형의 개수를 모드 곱한 후 속성 전체가 선택되지 않는 1가지의 경우를 제외한 값이 된다.

(3)트랜잭션별 검색 유형 데이터

트랜재션별 검색 유형 데이터는 한 명의 학생이 단일 과목의 다중 항목에 대해 여러 번의 트랜잭션을 수행할 경우 이를 통합된 트랜잭션으로 구분하여 연관 규칙을 발견할 때 필요하다. 트랜잭션별 검색 유형 데이터를 생성하기 위해 <표 7>의 학생간

검색 유형 테이블에서 <S_no, A_code> 두 개의 항목을 근간으로 하여 <표 10>의 트랜잭션별 검색 유형 테이블을 생성한다.

트랜잭션별 검색 유형 테이블를 사용하는 이유를 두 가지로 정리할 수 있다.

첫째는 트랜잭션별로 학생이 검색한 과목 검색 유형간에는 어떤 연관성이 있는지를 발견하기 위해서이다. 두번째는 속성간 검색 유형 데이터에서의 연관 발견은 단일 항목에서 선택할 경우, 연관 규칙을 적용할 수 없기 때문에 트랜잭션별 검색 유형 데이터를 사용함으로서 단일 항목에서만 검색 유형을 선택한 경우에도 연관 규칙을 적용할 수 있도록 하기 위해서이다.

4.2.3 연관 마이닝을 이용한 연관 정보생성 모듈

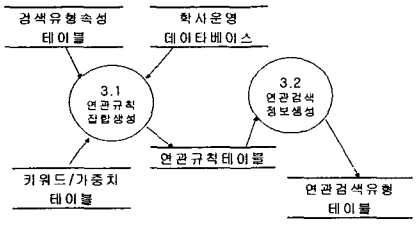
학생이 자주 검색하는 요구 조건들로 구성되어 있는 연관 검색 유형 데이터베이스는 일정 수준 이상의 트랜잭션이 발생되어야 연관 마이닝을 적용할 수 있기 때문에 다음과 같은 방법으로 검색 유형 데이터베이스를 구성하였다. 초기에는 발생된 트랜잭션이 없기 때문에 연관 검색정보 데이터베이스의 검색 적중률을 향상시키기 위하여 자주 검색될 것으로 생각되는 과목을 개설과목 테이블로부터 임의 값으로 등록시킨다.

(1)연관 데이터 생성 자료 흐름도

<표 11> 선정된 연관 규칙 테이블

S_no	S_optin	$Item_1$	$item_2$	---	$item_n$	T_name
순번	검색유형	첫 번째 속성키워드	두 번째 속성키워드		n번째 속성키워드	트랜잭션 테이블명

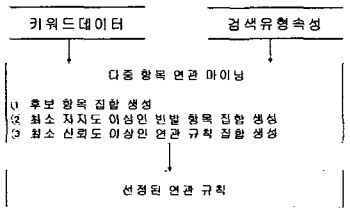
<그림 17>는 연관 마이닝 방법을 기반으로 하여 연관 검색 정보를 생성하기 위한 자료의 흐름도이다. 2장에서 선행연구에서의 연관 알고리즘을 사용하여 연관 규칙 집합을 생성하고 이에 해당하는 검색 유형별 트랜잭션 데이터를 생성한다.



<그림 17> 연관 검색정보 생성 자료흐름도(수준2)

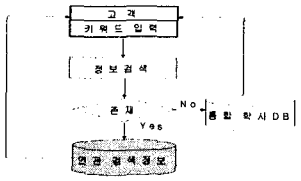
(2)연관 규칙 집합 생성

<그림 18>은 두 가지 형태의 검색유형 속성 데이터에서 연관 마이닝 방법을 적용하여 선정된 연관 규칙 집합을 생성하는 과정이다.



<그림 18> 선정된 연관 규칙 집합 생성

<표 11>은 빈발 항목 집합에서 연관 규칙 테이블을 생성할 때 최종적으로 최소 지지도와 신뢰도 이상을 가지는 트랜잭션 유형을 저장하는 선정된 연관 규칙 테이블이다. 선정된 연관 규칙 테이블에는 연관성이 있는 속성의 키워드를 보관하고 있기 때문에 이에 해당하는 실제 데이터를 학사운영 데이터베이스에서 가져와야 한다. 트랜잭션 테이블명은 전체 학사운영 데이터베이스에서 검색 유형에 해당하는 데이터를 보관하고 있는 테이블이다. <표 12>는 선정된 연관 규칙 집합의 예제로 S_k(1)은 선택 과목의 교수가중치가 설정되어 있음을 알 수 있고 검색 유형으로 이에 해당하는 데이터는 검색 유형 테이터 테이블 T1에 저장되어 있음을 알 수 있다.



<그림 19> 과목 검색 과정

(3)연관 검색 정보 데이터베이스 생성

<표 13>은 연관 규칙 집합으로 선정된 검색 유형별로 해당하는 검색 결과 데이터를 저장하기 위해 필요한 검색 유형 데이터 테이블의 구성이며 <표 14>는 이의 예제 테이블 데이터이다.

(4)연관 검색 유형에 의한 정보 검색

<그림 19>는 이러한 과목 검색 과정을 나타낸다.

```
Algorithm Serching_System
C_kind : 학생이 검색한 유형, S_kind(i) : i번째 검색 유형,  Sk_tbl(i) : I번째 검색 유형 데이터 테이블
Tot_tbl : 전체 과목 데이터 테이블,    n : 검색 유형 데이터 테이블 개수
for(i=1; i≠0; i++)
  if (C_kind=S_kind(i)) then Search((Sk_tbl(i))
  else Search(Tot_tbl)
  endif
end
```

<그림 20> 키워드 과목 검색 알고리즘

<표 12> 선정된 연관 규칙 테이블 예

순번	검색유형	다중항목에서 선택된 속성				트랜잭션 테이블명
		교수 가중치	학생선호 가중치	---	전공 가중치	
1	S_k(1)	●		---		T1
2	S_k(2)		●	---		T2
3	S_k(n)	●	●	---	●	Tn

<표 13> 연관 검색정보 데이터 테이블

S_no	Open_year	Open_term	Open_dept	Open_class	---	Open_code	S_optin
순번	개설년도	개설학기	개설학과	개설학년		과목코드	검색유형

<표 14> 연관 검색정보 데이터 테이블 예제

순번	개설년도	개설학기	개설학과	개설학년	이수구분	과목코드	검색유형
1	2001	2	컴퓨터정보	3	1	C1234	S_k(2)
2	2001	2	컴퓨터정보	3	2	C2345	S_k(2)
3	2001	2	컴퓨터정보	4	1	C3456	S_k(2)

<표 15> 키워드 데이터 테이블

S_no	Open_year	Open_term	Open_dept	Open_class	---	Open_code	S_result
순번	개설년도	개설학기	개설학과	개설학년		과목코드	점수

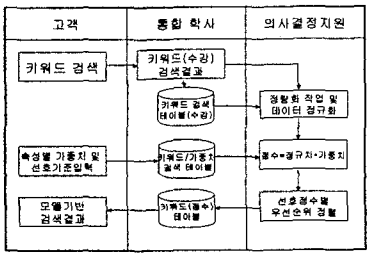
학생이 키워드로 검색한 결과는 <표 15>에 있는 키워드 검색 테이블에 저장한다. 이것은 다수의 키워드 검색 결과에 대해 학생 선호 점수 순위를 이용하여 학생에게 알맞은 과목을 우선적으로 제시하기 위해 필요하다. 점수(result) 속성은 학생 선호 기반의 검색 모듈에서 과목별로 학생의 선호도를 계산하여 선호 점수를 저장하기 위한 것이다.

<그림 20>는 <그림 13>의 과목 검색 과정에서 키워드 검색 결과를 제공하는 알고리즘이다. 학생이 검색한 유형이 연관 검색 정보 데이터 테이블에 있는 검색 유형과 동일한 것이 있으면 해당 테이블에서 검색 결과를 가져오고 그렇지 않으면 전체 교과목 데이터 테이블에서 검색 결과를 가져온다.

4.2.4 학생 선호 과목 의사결정지원 검색 모듈

(1)다속성 의사결정이 점수 모델 구성도

<그림 21>은 다속성 의사결정 방법에서 점수 모델의 처리 과정을 나타내고 있다. 수강신청에서 검색한 데이터 속성은 정성적 또는 정량적인 특성을 가지기 때문에 점수를 계산하기 위하여 정성적인 속성의 데이터는 정량화로 변환되어야 한다.



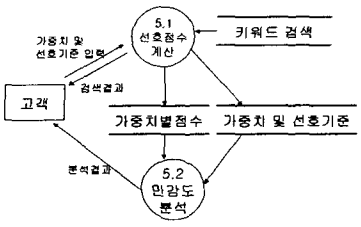
<그림 21> 점수 모델의 구성도

(2)학생 선호 과목 의사결정지원 자료 흐름도

<그림 22>은 학생 선호 과목 모델을 기반으로 하는 검색에 대한 자료 흐름도이다.

(3)선호 점수 계산 알고리즘

다속성 의사결정 모델에서 학생에게 적합한 의사결정을 지원하기 위해서는 여러 가지 모델들이 있는 데 본 연구에서는 점수 모델의 단순 가중치 방법을 가용한다. <그림 23>은 학생의 선호도에 따른 과목의 점수를 계산하는



<그림 22> 학생선호 모델 기반의 자료 흐름도(수준2)

다. (b)의 가중치 및 선호 기준 테이블은 점수계산에 사용될 학생이 입력한 선호 가중치와 선호기준을 저장하기 위한 테이블이다. (c)의 가중치별 점수 테이블은 민감도 분석에서 사용하기 위해 선호 점수 계산에서 나온 다수의 검색에 대한 점수를 저장하기 위한 테이블이다.

(5)선호도 분석

선호도 분석은 학생이 선호하는 속성에 대해 가중치의 반영 비율을 변화하면서 분석을 수행한다. 학생은 다양한 가중치를 제시함으로서 적합한 과목을 선정하는 데 도움을 받게 되며 이러한 분석 결과로 학생들의 선호도에 맞는 과목을 신설하거나 폐강할 수 있는 참고 자료로 활용할 수 있다. 그러나 단순한 수강 용이성만을 선호하는 경향으로 관련 학과의 전공 부실 및 나아가 교육의 질적 향상을 저해할 수 있는 경향도 분석하고 그 대책을 세울 수도 있다.

```
Algorithm jumsu_Computation

JUMSU(i) : 과목별 선호 점수, D(i,j) : 레코드의 항목 속성값,  ND(i,j) : 각 속성의 정규치
PRE(j) : 각 속성의 선호기준(1:이윤속성, 0:비용속성)
Max[Dj(i,j)] : 각 속성의 최대값,          Min[Dj(i,j)] : 각 속성의 최소값

W(j) : 각 항목의 가중치 (0≤W(j)≤1,   Σ_{j=1}^{m} W(j) = 1)
for(i=1, n; i++) do begin                /* i번째 레코드
 for(j=1, m; j++) do begin               /* I번째 레코드의 j번째 항목
   if (PRE(j)=1) then  ND(i,j) = D(i,j) / Max[Dj(i,j)]   /* 이윤속성
   else  ND(i,j) = Min[Dj(i,j)] / D(i,j)    /* 비용속성
   endif
    JUMSU(i) = JUMSU(i) + W(j) * ND(i,j)
 end
end
```

<그림 23> 학생 선호도에 따른 점수 계산 알고리즘

(4)모델베이스에 필요한 테이블

학생 선호 기반의 모델 관리 시스템에 저장되고 모델베이스에 필요한 테이블은 <표 16>과 같다. <표 16>의 (a)의 키워드 검색 테이블은 학생이 키워드 입력으로 검색한 데이터를 저장하는 테이블이며 선호 점수 계산 업무에서 입력 데이터로 사용된다. 여기서 점수(jumsu) 필드의 값은 학생이 가중치와 선호 기준을 입력하였을 때 계산된

5. 결 론

<표 16> 모델 테이블 구조

(a) 키워드 데이터 검색 테이블

S_no	Open_year	Open_term	Open_dept	Open_class	---	Open_code	S_result
순번	개설년도	개설학기	개설학과	개설학년		과목코드	점수

(b) 가중치 및 선호 기준 테이블

W_no	W_code	W_prof_w	W_prof_o	W_dept_w	W_dept_o	---
가중치 순번	가중치과목코드	교수 가중치	교수 선호기준	학과 가중치	학과 선호기준	---

(c) 가중치별 점수 테이블

W_no	W_code	jumsu
가중치 순번	가중치 과목코드	점수

보다 현실적인 정보를 제공할 수 있는 데이터 마이닝에 의한 마트의 성장과 연관 검색 정보를 이용하여 연관 알고리즘(Algorithm)에 의한 정보의 축척과 질의 정보를 효율적으로 제공하기 위한 방안으로 본 논문에서는 연관 마이닝 방법을 적용하여 연관 검색 유형 데이터베이스를 생성하여 검색 속도를 빠르게 할 수 있도록 데이터마트를 생성하고 학생의 속성별 가중치와 선호기준을 입력받아 선호 점수를 계산하여 점수가 높은 상품을 우선적으로 검색할 수 있도록 데이터베이스캐시화로 기존 연관 알고리즘에서 사용한 단일 항목 입력 데이터 구조를 확장하여 다중 항목 연관 알고리즘(Multiple Item Association Mining : MIAM)을 이용하여 생성된 연관 검색 유형 테이블을 데이터베이스캐시를 설계하였다. 그 결과 동일한 알고리즘에서도 데이터베이스 캐시 시스템을 적용한 시스템의 질의처리 수행속도의 우수성을 설계함으로써 효율적인 웹 서버의 기능을 수행할 수 있음과 동시에 데이터베이스 캐싱의 주요 이점인 효율성 증대, 속도 향상, 비용절감의 기대효과를 얻을 수 있다. 이러한 기능은 실제적 적용의 측면에서 보면 다중 항목 연관 알고리즘을 이용하여 다중 과목별, 학생 분류별로 선호하는 상품 정보를 제공할 수 있으므로 과목 선택 기준에 유용할 것으로 생각된다.

향후 본 연구의 가상대학에서 수강신청 과목 선택의 검색 시스템 적용에 있어서 제시한 설계 시스템을 구현하여 실제 가상대학에서 운영하여 수집한 실제 데이터와 실험적인 결과에서 얻은 데이터와 비교 분석해 보는 과정이 필요하며, 연관 마이닝 알고리즘의 다중항목 적용을 위한 데이타베이스캐시 설계의 간편화와 업무 활용도 증가를 위한 구체적인 보다 다양한 교과목 편성과 학생 분류별로 과목의 연관성을 추출하여 수강과목 선택의 편리성을 지원하는 과목 검색 시스템에 대한 연구가 필요하다.

참 고 문 헌

[1] 황현숙, 연관 마이닝 방법을 이용한 상품 검색 의사결정지원시스템 설계 및 구현, 부경대 대학원 박사학위 논문, 2001년 2월

[2] Agrawal, R., Imielinski, T. and Swami, A., Mining Association Rules between Sets of Items in Large Database, Proceedings of ACM SIGMOD Conference on Management of Data, 1993, pp.207-216.

[3] Agrawal, R. and Srikant, R., Fast algorithms for mining association rules, Proceedings of the 20th VLDB Conference, 1994.

[4] Berson, A. and Smith, S., Data Warehousing Data Mining and OLAP, Mcgraw-Hill, 1997.

[5] Brin, S., Motwani, R., Ullman, J.D. and Tsur, S., Dynamic Itemset Counting and Implication Rules for Market Basket Data, Proceedings of ACM SIGMOD Conference on Management of Data, 1997, pp.255-264.

[6] Houtsma, M. and Swami, A., Set-oriented Mining of Association Rules, Report RJ 9567, IBM Almaden Research Center, 1993.

[7] Lee, S., Lee J. and Lee, K., Customized Purchase Supporting Expert System: UNIK-SES, Expert Systems with Application, Vol. 11, No. 4, 1996.

[8] Andriaans, P. and Zantinge, D., Data Mining, Addison-Wesley, 1996.

[9] Fayyad, U.M., Gregory P., Padhraic S. and Ramasamy, U., Advances in Knowledge Discovery and Data Mining, MIT press, 1995.

[10] Sarawagi, S., Thomas, S. and Agrawal, R., Integrating Association Rule Mining with Relational Dtabase Systems : Alternatives and Implications. Dta Mining and Knowledge Discovery, Vol. 4, No. 2, 2000, pp. 80-125.

[11] Mearhos, J., Rothman, M. and Vivers, M., Applying Data Mining Techniques to a Healthy Insurance Information System., Proceedings of the 22nd International Conference on Very Large Databases, 1996.

[12] Jeon, T.G., Hwang, H.S., Kim, C.S., Shim, K.B. and Shim, D.S., A Study on the Association Mining Algorithm for Intrusion Detection, Proceedings of International Conference on EALPIIT, 2000, pp.26-31.

[13] Toivonan, H., Sampling Large Databases for Association Rules, Proceedings of the 22nd VLDB Conference, 1996.

[14] Agrawal R. and Shim, K., Developing Tightly-Coupled Data Mining Applications on a Relational Database System, Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining, 1996.

[15] Srikant, R., Agrawal, R., Mining Quantitative Association Rules in Large Relational Tables, Proceedings ACM SIGMOD, 1996.

[16] Tsur, D., Abiteboul, S., Clifton, C., Motwani, R. and Nestorov, S., Query Flocks : A

Generalization of Association-Rule Mining, SIGMOD, 1998.
[17] Thomas, S. and Sarawagi, S., Mining Generalized Association Rules Rules and Sequential Patterns Using SQL queries, *Proceedings of the 4th International Conference* on Knowledge Discovery in Database and data Mining, 1998.
[18] Imielinski, T., Virmani, A., MSQL : A Query Language for Database Mining, Data Mining and Knowledge Discovery, Vol. 3, No. 4, 1999, pp. 373-408.
[19] Blanning, R.W, Model Management Systems : An Overview, Decision Support Systems, Vol. 9, No. 1, 1993, pp. 19-37.
[20] Banerjee, S. and Basu. A, Model type selection in an integrated DSS environment, Decision Support Systems, Vol. 9, No. 1, 1993, pp. 75-89.
[21] Craig, S., Sergey, B. and Rajeev, M., Beyond Market Baskets : Generalizing Associations Rules to Dependence Rules, Data Mining and Knowledge Discovery, Vol. 2, No.1, 1998, pp. 39-68.

사 재 학 (Jae-Hak Sa)
e-mail: jhsa@kimcheon.ac.kr
1990년 경일대 전자계산학과 졸업(공학사)
1993년 대구대 산업정보대학원 산업정보학과 전산공학전공(공학석사)
2000년 대구대 대학원 컴퓨터정보공학과 전산공학전공 (박사과정 수료)
1984~1987 정우정보산업(주)시스템개발부 근무
1987~1988 (주)NCS & 경일 개발부 근무
1988~1990 포항강재공업(주) 전산팀 근무
1990~1991 안동대학교 전자계산소 근무
1991~1992 대구경북개발연구원 전산실 근무
1992~1996 책임기업(주) 전산부 근무
1996~1998 상주적십자병원 의료정보실 근무
1998~1999 대구미래대학 전산과 전임강사
1999~2001 同대학 멀티미디어정보과학과전임강사
1998~2000 同 미래정보센터 전산과장
2001.9~ 김천대학 컴퓨터정보처리계열 초빙교수

관심분야 : DB, 멀티미디어, SW공학, 언어

남 인 길 (In-Gil Nam)
e-mail:ignam@taegu.ac.kr
1978년 경북대학교 전자공학과 (공학사)
1981년 영남대학교 대학원 전자공학과 계산기전공(공학석사)
1992년 경북대학교 대학원 전자공학과 전산공학전공(공학박사)
1978년~1980년 대구은행 전산부
1980년~1990년 경북산업대학 전자계산학과 부교수
1996년~1997년 미국 루이지애나 주립대학 교환교수
1990년~현재 대구대학교 컴퓨터정보공학부 교수
2000년~2001 대구대학교 학생처장
2001년~현재 대구대학교 사무처장

관심분야 데이터베이스, GIS, 이동컴퓨팅