

유비쿼터스 환경에서 복합 상품 구성을 위한 지능형 여행 정보 시스템에 대한 연구 - 의미론적 웹 서비스 중심

이현정
성균관대학교 경영대학 경영연구소
(hjlee5249@yahoo.com)

손미애
성균관대학교 공과대학 시스템경영공학과
(myesohn@skku.edu)

본 논문에서는 유비쿼터스 환경에서 유동 소비자에 대한 서비스 이용의 유동성을 보장할 수 있는 지능형 여행정보서비스를 제안하고자 한다. 본 시스템은 유비쿼터스 환경에서 여행자가 필요로 하는 다양한 정보 획득을 돕기 위해 단일상품 묶음을 통한 복합상품 구성과 시맨틱 웹 서비스에 기반한 정보의 구조적 표현을 목적으로 설계되었다. 여행정보시스템에 연관된 상품들은 호텔, 항공사, 자동차대여, 여행지 정보, 음식점 및 Business Trip 등으로 여행자의 요구 및 상황의 변화에 맞게 연관상품의 복합상품구성을 위한 지능형 여행 최적 정보시스템의 연구가 요구된다. 이를 위해 본 연구에서는 여행서비스 지원을 위해, 첫째 여행자의 요구에 따른 단일상품의 복합상품화를 위한 방법론(Product bundling and modification)을 제안하였다. 둘째, 복합상품을 구성하기 위해 개별 정보시스템에 이질적인 형태로 저장되어있는 단일상품에 대한 정보를 공유하기 위해 시맨틱 웹 기반 정보의 구조적 표현에 관하여 제안한다. 이에 본 논문에서는 OWL 기반의 온톨로지를 구축하였다. 따라서 온톨로지 기반의 여행정보시스템은 향후 시맨틱 웹서비스 기반의 지능형 여행정보시스템 구축을 위한 초석으로 활용될 것이다.

논문접수일 : 2005년 11월

게재확정일 : 2006년 6월

교신저자 : 이현정

1. 서론

정보통신 기술의 발전은 사용자들이 네트워크가 연결된 곳이라면 언제, 어디서나 원하는 정보를 제공받을 수 있도록 하는 것이 가능하게 하였다. 그러나 사용자가 서비스를 받기 위해서는 반드시 네트워크가 연결된 곳으로 이동 해야 하는 한계가 있다. 즉 네트워크가 가능한 위치로의 이동이 곤란하다면 서비스 받는 것이 쉽지 않음을 의미한다. 따라서 사용자의 유동성 지원을 위한

연구들이 e-Business 분야에서 수행되었으며 [9, 16], 이러한 노력이 집중된 분야가 유비쿼터스 또는 pervasive 컴퓨팅 이다. 즉 유비쿼터스 환경에서는 사용자가 언제 어디에 위치하느냐와 관계없이 자신이 필요로 하는 정보를 제공받을 수 있으며, 나아가 유동 소비자를 위한 정보의 유동서비스 제공이 가능해진다.

이러한 유동성 보장은 여행자의 경우에 특히 중요한 문제로 대두된다. 유동성이 큰 사용자 중의 하나인 여행자는 여행 중에 예기치 않은 계획

변경 상황에 노출이 되기 쉽다. 여행자는 자신이 처한 상황 변화에 대처하기 위해, 유동성 있는 정보시스템에 접속하여 자신의 변경된 계획에 맞춰 수정된 숙박, 교통, 여행지의 날씨 및 기타 필요한 여행정보를 종합적으로 제공받기를 기대할 것이다. 이러한 기대를 충족하는 서비스를 제공하기 위해, 본 논문에서는 유비쿼터스 환경하에서 여행자가 다양한 정보를 손쉽게 요약하여 제공 받을 수 있도록 시멘틱 웹서비스 기반 단일상품의 묶음에 의한 복합상품 구성을 위한 지능형 여행 정보 시스템을 제안하였다.

여행상품의 특성은 단일상품의 결합에 의한 복합상품 구성이 가능하다는 것과 복합 상품 구성을 위해 다양한 정보자원의 공유가 필요하다는 것이다. 여행상품의 주 사용자는 유동 소비자로서, 여행상황의 변화에 따라 요구하는 정보 및 상품의 유형이 실시간으로 변화할 수 있다는 특징을 가지고 있다. 이는 소비자의 유동성이 커지는 것에 비례하여 상품의 유동성 역시 상당히 증가할 수 있음을 의미한다. 따라서 유비쿼터스 환경에서 여행자가 언제 어디서나 여행상품과 관련 정보를 적절히 제안 받을 수 있는 시스템의 필요성은 상당히 크다.

본 논문에서 제시하는 여행정보시스템은 여행상품의 복합상품화를 위한 묶음상품 방법론과 구조적인 정보 처리를 위해 요구되는 온톨로지 구축 및 이의 활용과 관련된 연구를 중심으로 설계하고 방법론을 제안하였다.

본 연구에서는 제2장에서 관련연구를 살펴보고, 제3장에서는 복합상품 구성에 위한 방법론과 복합상품 구성을 위한 구조적 서비스를 논의 한다. 제4장에서는 상품의 구조적 표현과 복합상품 구성 예제를 살펴보고 구현을 통한 제시된 시스템의 효율성을 살펴본다, 마지막으로 제5장에서는 결론 및 향후 연구방향을 살펴보고자 한다.

2. 관련연구

2.1 상품 검색, 묶음과 추천

유비쿼터스 환경에서는 소비자가 언제 어디서나 자신이 원하는 서비스를 제공 받을 수 있다. 그러나 다량의 다양한 형태의 정보제공은 사용자로 하여금 시시각각으로 수신된 정보를 활용하여 의사결정을 하도록 강요함으로써 사용자의 부담을 가중시키는 부작용을 야기할 수 있다 [8, 14]. 특히 본 논문에서 고려하고 있는 여행자의 경우 유동성이 큰 만큼 예상할 수 없는 상황 변화에 접할 수 있으며, 이러한 변화는 여행자의 의지와 상관없이 여행 일정 등의 변경을 야기할 수도 있다[19].

따라서 유비쿼터스 환경에서는 여행상품은 수시로 발생할 수 있는 변동사항을 즉각적으로 반영한 상품의 조합이 가능해야 하며, 이렇게 조합된 복합상품은 추천(recommendation)을 통해 용이한 방법으로 사용자에게 제안되어야 한다[5, 19]. 기존의 상품추천 시스템은 주로 단일 상품 추천을 위해 사용자의 요구사항 및 성향에 분석 등에 초점을 두고 있었다[1, 12, 15, 17, 20]. 본 연구에서는 기존의 단일상품의 추천과 달리 사용자의 요구사항 따른 단일 상품들을 복합 상품화하는 방법론에 초점을 두고자 한다.

복합상품을 구성하는데 활용할 수 있는 방법 중의 하나가 재구성(reconfiguration)이다. 재구성은 소비자의 변화된 요구에 맞춰 기존의 상품의 항목 값에 변화를 주는 것으로써, 새로운 제품을 구성하는 것보다 상품검색에 걸리는 노력과 시간을 절감할 수 있다는 장점을 가지고 있다[4, 12, 15].

이외에도 상품을 검색하는 다양한 방법들이 연구되어 왔으며 상품을 구성하는 구성요소들 간의 복잡도가 증가하면서 검색의 어려움도 따라서 증

가하였다. 근래에는 의미(semantic)를 이용한 지능형 검색 방법도 많이 연구되고 있다[21]. 의미기반 검색에서는 사용자가 다양한 쿼리를 만들 수 있고, 이를 시맨틱 웹을 이용해 사용자의 요구에 맞는 상품검색을 지원하는 시스템 등이 있다.

그러나 단순히 상품 검색이나 재구성하는 방법 [12] 만으로는 고객의 요구사항에 맞는 복합상품을 제안하는 것이 쉽지 않다. 특히, 여행상품과 같이 상품의 유동성이 크고 단일 상품의 묶음으로 복합상품 등으로의 구성에 관한 연구가 요구된다.

2.2 의미 웹 기반 프로세스

여행상품의 복합화 과정은 다양한 정보의 통합 과정이다. 예를 들어, 학회에 참석하고자 하는 고객을 여행 복합상품에는 학회정보, 항공기 정보 및 호텔정보 등이 묶음(bundling)되어 제공되어야 할 것이다. 그러나 묶음 되는 정보들은 서로 다른 서버에 이질적인 형식의 데이터로 저장되어 있는 경우가 대부분이다. 이들로부터 목표 복합상품의 묶음에 필요한 정보를 식별하고 추출하기 위해서는, 여행상품에 관련된 온톨로지의 구축이 선행되어야 한다. 온톨로지는 “특정 집단의 공유된 개념 (shared conceptualization)에 대한 정형화되고 명시적인 명세(formal and explicit specification)”로 정의 [10]되며, 웹 상에서 이질적인 개체들이 정보를 공유하기 위해서는 반드시 필요하다[2, 3, 19]. 구축된 온톨로지는 logic 또는 규칙과 결합되어 가치 있는 지식 (knowledge)을 생성하는데 활용될 수 있다. 온톨로지의 발전은 웹서비스의 진화에도 큰 기여를 하고 있다. 인터넷상에서 애플리케이션과 애플리케이션간의 상호 운용을 지원하기 위해 설계된 아키텍처인 웹서비스[24]를 자동화하기 위한 언어인 DAML-S(DARPA Agent Markup Language for Services)와 OWL-S[22,

23] 등이 등장했으며, 이들 언어의 등장으로 온톨로지 기반의 시맨틱 웹서비스의 시대가 본격적으로 열리게 되었다. 웹서비스의 SOA(Service Oriented Architecture)를 이용해 복합상품을 구성할 경우, 서비스 제공자는 복합상품 구성할 때 지켜야 할 제약조건을 준수해야 한다. 이러한 제약조건이 인터넷상에서 원활하게 교환될 수 있는 방법이 요구된다. 또한 웹서비스 아키텍처에서 지능적인 여행상품의 제안을 위해서는 지능형 에이전트의 도입이 필요하며, 에이전트들이 상호 데이터와 정보를 교환하거나 참조 또는 활용을 위해서는 온톨로지의 구축이 요구된다.

또한 복합여행상품의 구성과정에서도 다양한 제약조건이 규칙의 형태로 영향을 미치게 된다. 이들 규칙들은 명시적인 형태 또는 암묵적인 형태로 존재한다. 암묵적인 형태로 비구조적인 문서에 내재된 경우에는 XRM을 사용함으로써 형식화할 수 있다[13]. 본 논문에서는 복합여행 상품의 구성과정에 영향을 미치는 규칙을 형식화하기 위해 XRM을 채택하였다.

3. 복합상품 구성 방법론 - Product Bundling and Modification (PBM)

본 절에서는 여행상품 추천을 위해 묶음 상품을 위한 방법론인 *Product Bundling and Modification (PBM)* 알고리즘을 제안하고, 단일상품으로 복합상품을 구성하기 위한 정보자원의 공유를 지원하는 시맨틱 웹에 기반한 구조적 정보 표현에 대해 살펴보고자 한다.

3.1 복합상품 구성 프로세스

여행상품은 단일 상품과 서비스가 고객의 요구

사항에 맞게 결합됨으로써 복합상품화 되는 특성을 가지고 있다. 여행상품에서의 단일상품은 항공상품, 숙박상품, 교통상품, 여행정보상품 등을 의미하며, 이들은 고객의 요구사항에 따라 묶음으로 제동되게 된다. 예를 들어, 출장을 계획하고 있는 고객을 위해 복합상품의 구성이 가능하며, 이외에도 여행자의 일정에 따라 축제상품, 이벤트상품 등과의 연계도 상품 묶음을 통해 가능하다. 또한 여행 구성원 및 연령에 따라 다양한 상품을 구성할 수 있으며, 항공기 운항시간의 지연 및 연착 등에 따른 여행자의 여정의 변화에 따른 상품의 재조합, 날씨에 따른 여행상품 선택의 변화 등이 가능하다. 따라서 다양한 환경 조건의 변화에 따른 다양한 상품의 조합을 위한 방법론을 제안하고자 한다.

PBM에서 단일상품(Component)은 복합상품을 구성하는 기본 단위가 되며, 묶음 상품 구성(Product bundling processing) 절차는 고객의 요구사항에 따른 단일상품 선정(components selection) 절차와 선정된 단일상품의 복합구성 및 수정 절차로 구성된다. 즉 고객의 요구사항에 따라 연관된 변수들간의 연관관계에 의한 일치성(Consistency check)과 충돌문제해결(Conflict resolution) 등이 요구된다. 따라서 제안된 PBM의 방법론의 과정을 단계별로 살펴보면 다음과 같다.

Product Bundling and Modification Algorithm

Step 0 : Customer's requirements

Step 0에서는 고객의 요구사항에 따라 복합상품 구성을 위한 단일상품을 추출한다.

요구사항 (Customer's requirements) R 은 다음과 같다.

$$R = \{r_i \mid r_i \text{ is requirement, } 1 \leq i \leq p\}$$

복합상품 구성을 위한 단일상품들은 다음과 같

이 표현된다.

$$C = \{C_i \mid C_i \text{ is component } i, 1 \leq i \leq l\}$$

Step 1 : Components Selection

추출된 단일상품들에 존재하는 전역변수(global variable) 들을 정한다. 각 단일 상품 내에서 전역 변수를 제외한 나머지 변수들은 지역변수로 정의한다.

$$G_{var} = \{g_{v,i} \mid g_{v,i} \in C_1 \text{ and } g_{v,i} \in C_2, 1 \leq i \leq m\} : \text{전역변수}$$

$$L_{var-cl} = \{l_{v,i} \mid l_{v,i} \in C_1 \text{ and } l_{v,i} \notin C_2, 1 \leq i \leq n\} : \text{지역변수}$$

단일상품은 연관 있는 전역변수의 속성과 연관 관계를 계승한다.

$$G_{rc} = \{g_{rc,i} \mid g_{rc,i} \in C_1 \text{ and } g_{rc,i} \in C_2, 1 \leq i \leq q\}$$

$$L_{rc-cl} = \{l_{rc,i} \mid l_{rc,i} \in C_1 \text{ and } l_{rc,i} \notin C_2, 1 \leq i \leq t\}$$

복합상품 구성에서 단일상품에의 변수들간의 연관관계는 제약조건(constraints)과 규칙(rules) 형태로 복합상품에 계승된다. 따라서 지역변수들간의 제약조건과 규칙들은 복합상품에서도 단일상품에서와 마찬가지로 유지되며, 전역변수들 사이의 그리고 전역변수와 지역변수들 간의 새로운 연관관계가 형성된다.

Step 2 : Composition of Components

Step 2-1 : Value Change

선택된 단일 상품의 변수의 값이 고객의 요구사항과 일치하지 않으면 해당 변수의 값을 고객의 요구사항 값으로 변환한다.

$$\text{If } var.required_value \neq var.selected_value \\ valuevar \leftarrow var.required_value$$

Step 2-2 : Consistency Check

변수의 값이 변화된 항목들간의 일치성은 다음

과 같은 하위 단계를 거쳐 조사한다.

- 1) 전역변수들간의 충돌이 발생한 변수들을 추출한다.
- 2) 전역변수와 지역변수 사이의 충돌이 발생한 변수들을 추출한다.
- 3) 지역변수 내에서 충돌이 발생한 변수들을 추출한다.

Step 2-3 : Components or optional variable addition

새로운 단일상품 또는 변수가 복합상품에 동적으로 추가될 수 있다. 추가될 단일상품을 추출하고 묶음 될 복합상품의 변수들과의 연관관계에 따라 전역변수와 지역변수를 결정한다.

Step 3 : Conflict Resolution

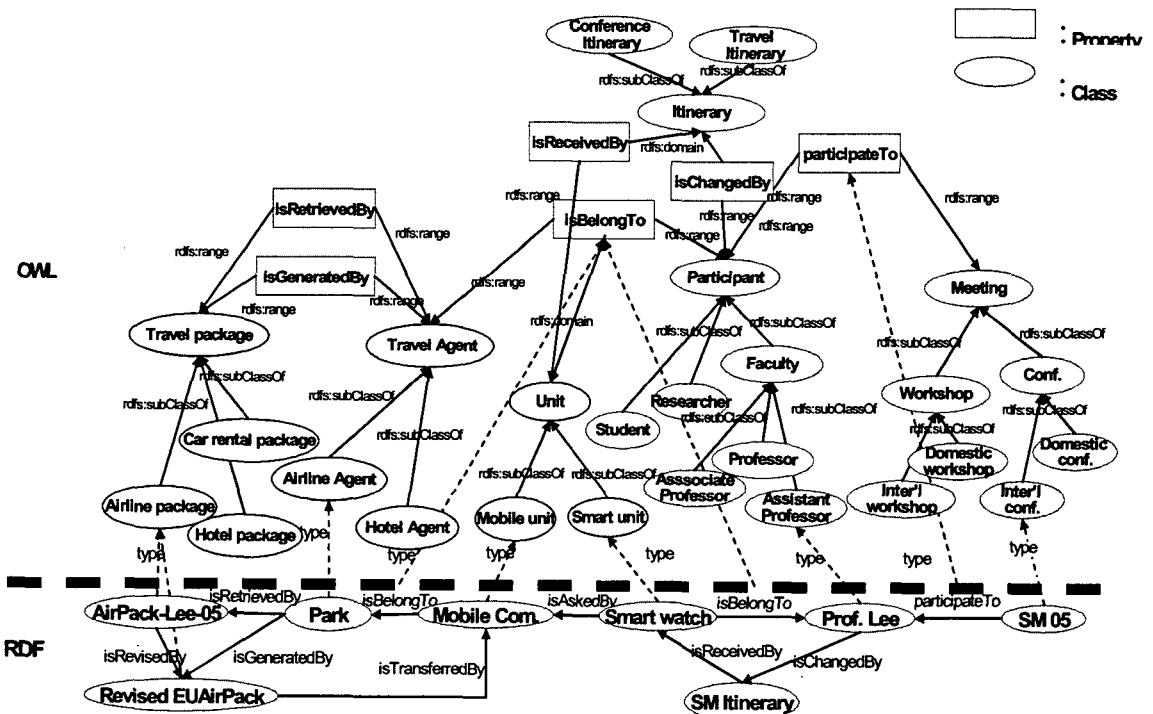
1) 상충해결을 위한 변수간의 우선순위(Variable ordering)는 다음과 같다.

즉, 변수들간의 Confliction이 발생하면 고객의 선호도가 낮은 변수의 값을 변경한다[12]. 지역변수와 전역변수간의 충돌이 발생하면, 선호도가 같은 경우는 지역변수의 값을 바꾼다.

2) 상충해결을 위한 전파 (propagation)의 시작점(seed selection)은 다음과 같다.

본 연구는 변화된 값을 가진 변수들이 충돌해결(Conflict resolution)을 위한 전파의 다중 시작점 (Multiple seeds selection)이 된다[12].

이때 전역변수는 복합상품의 상충해결을 위한 전파의 시작점이 되며 지역변수는 단일상품내의



[그림 1] 여행일정 변경을 위한 부분 온톨로지

상충해결을 위한 전파의 시작점이 된다.

3.2 복합 상품 묶음을 위한 구조적 서비스

여행상품은 분산된 정보시스템들과의 연계와 밀접한 관련을 가진다. 따라서 분산된 정보시스템들간의 정보공유를 위해 웹의 데이터교환 표준을 이용한 구조적 정보 표현 및 처리(Structured information processing)가 중요하다.

여행자에게 본 장에서는 단일 상품의 묶음을 통한 복합상품 구성을 위한 구조적 정보 표현을 제안한다.

단일상품 묶음을 통한 복합 상품을 제안하기 위해서는 다수의 관련 정보시스템간의 연계가 요구된다[19]. 개별 정보시스템은 하나 또는 복수개의 단일 상품들에 대한 정보를 내재하고 있다 이들 개별 정보시스템은 단일 상품을 서비스하는 기업들이 독자적으로 개발하여 운영하고 있는 시스템이다. 이들 시스템이 복합상품의 대상이 되는 단일 상품을 내재하고 있는 타 정보시스템과 데이터 또는 서비스정보를 교환하기 위해서는 웹서비스와 같은 아키텍처의 도입이 필요하다.

다음과 같은 시나리오를 생각해 보자. 유럽의 학회에 참여하고 있는 이교수는 학회의 참여자인 김교수의 제안으로 이탈리아를 방문하게 되었다. 갑작스런 일정의 추가로 이교수는 호텔과 항공권의 예약이 필요하게 되었다. 이교수의 에이전트는 이러한 상황 변경에 대응하기 위해 호텔과 항공사에 에이전트에 접속하여 새로운 상품을 구성하게 된다. [그림 1]은 이교수의 문제를 해결하기 위한 온톨로지 의미망 (semantic net)으로 도식화 한 것이다.

[그림 1]에서 타원은 클래스를 사각형은 프로퍼티를 나타낸다. 점선 아래 부분인 RDF는 국제학술대회인 'SM05'에 참석한 이교수의 일정과 관련된 인스턴스이고, 점선 위부분은 RDF에서 사용한

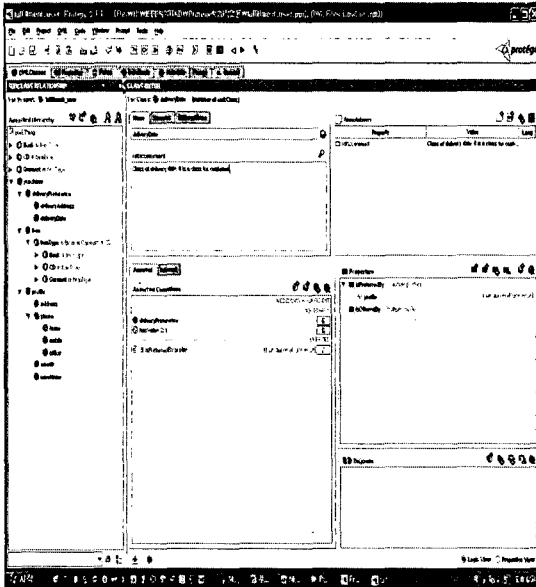
단어들의 관계를 일반화하고 있는 OWL이다. 또한 [그림 1]의 의미망을 웹 온톨로지 언어인 OWL로 표현하기 위해 Protégé 적용하였다. [그림 2]와 <표 1>은 Protégé 예제 화면과 OWL로 표현된 문법의 예이다.

<표 1> OWL로 표현한 여행상품관련 부분 온톨로지

```

<owl:Ontology rdf:about="Travel Package"/>
  <owl:Class rdf:ID="SmartWatch">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="SmartUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="IntenationalConference">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Conference"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Conference">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Meeting"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="AirlineAgent">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="TravelAgent"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#TravelAgent">
    <rdfs:subClassOf rdf:resource="#Agent"/>
  </owl:Class>
  <rdfs:Property rdf:ID="isBelongTo">
    <rdfs:domain rdf:resource="#Participant"/>
    <rdfs:range rdf:resource="#Unit"/>
  </rdfs:Property>

```



[그림 2] Protégé로 표현한 OWL 온톨로지

본 연구에서는 이상의 온톨로지를 기반으로 하여 단일여행상품을 복합상품으로 구성하기 위한 구조적 표현방법을 제안한다.

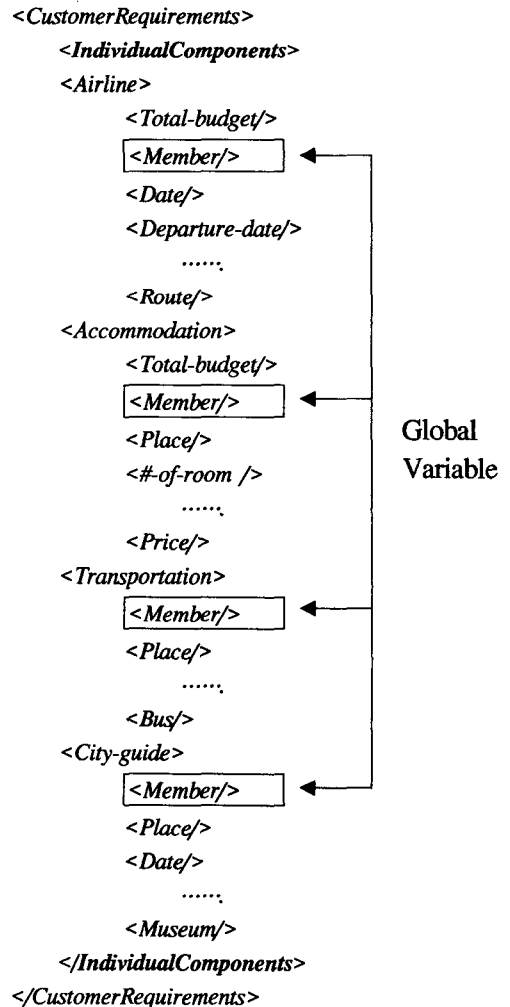
4. 복합 상품 구성의 구현 및 분석

본 절에서는 상품의 구조적 표현과 복합상품 구성예제를 살펴보고 실험을 통해 본 연구에서 제안한 시맨틱 웹기반 PBM의 효율성을 논의하기로 한다.

4.1 상품구성을 위한 구조적 표현

고객의 요구사항은 복합상품 구성을 위한 단일 상품들로 표현되어 의미 기반 웹 서비스를 이용하여 필요로 하는 단일상품에 대한 정보 추출에 기본이 된다. 추출된 단일상품이 복합상품을 이루기

위해 단일상품내의 변수는 전역변수와 지역변수로 구분된다. 전역변수는 복합상품을 구성하는 둘 이상의 단일 상품들이 공동으로 만족해야 하는 변수들이다. 이의 표현을 위해 구조적 정보 처리 (Structured information processing)가 되며, 이를 위해 XML로 표현하면 다음과 같다.



위의 단일상품들 중 전역변수와 지역변수를 식별해 복합 상품을 구성하면 아래와 같다

```

<CompositeComponents>
  <Global>
    <Member/>
    <Date/>
    <Place/>
    <Total Budget/>
  </Global>
  <Local>
    <Airline>
      <Air-departure-date/>
      .....
      <Air-price/>
    </Airline>
    <Accommodation>
      <acc-departure-date/>
      .....
      <acc-room-type/>
    </Accommodation>
    <Transportation>
      <trans-rental/>
      .....
      <trans-subway/>
    </Transportation>
    <City-Guide>
      <city-museum/>
      .....
      <city-stadium/>
    </City-Guide>
  </Local>
</CompositeComponents>

```

전역변수는 연관된 단일상품에게 그 속성들과 연관관계가 계승된다. 따라서 추출된 전역변수와 지역변수들 간의 제약조건과 규칙들은 기존의 단일상품 내에서의 연관관계를 기본으로 하여 구성된다. 또한 기존의 변수들간의 연관관계를 복합상품에서 활용하기 위해서는 구조적인 정보 표현의 예는 다음과 같다.

Airline

```

IF (Airline IS 단일상품)
Then (Air-price LE Total-budget)
Accommodation
IF (Accommodation IS 단일상품)
Then (Acc-Price LE Total-budget)

```

4.2 PBM 알고리즘에 따른 복합상품구성 예제

PBM 단계에 따른 복합상품 구성예제를 살펴보면 다음과 같다.

Step 0: Customer requirements

고객의 요구사항에 따라 복합상품 구성을 위한 단일상품을 정의 및 추출한다. 본 예에서는 단일상품으로 Airline, Accommodation, Transportation 및 City-guide가 선정되었다.

Step 1: Components Selection

단일상품들로부터 복합상품 구성을 위한 전역 변수들은 Total-Budget, Member, Date 와 Place 이다. Total-budget은 단일상품 Airline, Accommodation과, Member는 Airline, Accommodation, Transportation와, Date는 Airline, City-guide와 Place는 Accommodation, Transportation 및 City-guide와 연관되어 있다. 각 단일상품 Airline, Accommodation, Transportation, City-Guide는 하위 독립 변수들을 가지므로 독립된 단일상품으로 분류되며, 각 단일상품들은 아래와 같은 변수들을 가진다.

Components {Airline, Accommodation, Transportation, City-guide}

Global variables {Member, Date, Place, Total-Budget}

Components 1 : Airline

<Local variable>

Must variables {air-departure-date, air-arrival-date}

Optional variables {Total-budget, air-dinner, air-class, air-seat-number, air-route, air-price}

</Local variable>

Components 2 : Accommodation

<Local variable>

Must variables {*acc-departure-date*, *acc-arrival-date*}

Optional variables {*acc-#-of-room*, *acc-room-type*, *acc-price*, *acc-place*}

</Local variable>

Components 3 : Transportation

<Local variable>

Must variable { }

Optional variables {*trans-rental*, *trans-bus*, *trans-taxi*, *trans-train*, *trans-subway*}

</Local variable>

Components 4 : City-Guide

<Local variable>

Must variables { }

Optional variables {*city-museum*, *city-castle*, *city-tower*, *city-square*, *city-stadium*}

</Local variable>

<표 2>는 고객의 요구사항에 따른 복합상품과 각 변수들을 예시한다.

<표 2> Global Components to composite Traveling Products

Global Components	Variables (Preference)	Values
Member	Member Type (1.0)	Single/ Double/ Family
	#-of-member (1.0)	#(n)
Date	Departure-Date (1.0)	mm-dd-yy
	Arrival-Date (1.0)	mm-dd-yy
Place	Location (1.0)	Location
	Activity (1.0)	In / Out
Total-budget	Total Budget (0.8)	\$

<표 3>은 단일상품인 Airline과 그 변수들을 예시한다.

<표 3> Individual Component Airline to composite Traveling Products

Individual Components	Variables (Preference)	Values
Airline	Departure-date (1.0)	mm-dd-yy
	Arrival-date (1.0)	mm-dd-yy
	Dinner (0.8)	Regular/ Kids/ Special
	Class (0.6)	Economy /Business /First
	Seat number (0.4)	n
	Route (0.8)	Via / Direct
	Price (0.8)	\$

<표 4>는 단일상품인 Accommodation과 그 변수들을 예시한다.

<표 4> Individual Component Accommodation to composite Traveling Products

Individual Components	Variables (Preference)	Values
Accommodation	Departure-date (0.8)	mm-dd-yy
	Arrival-date (0.8)	mm-dd-yy
	#-of-room (0.6)	#(n)
	Room-type (0.4)	Single/ Double/ Family
	Price (0.6)	\$
	Place (0.8)	Location

<표 5>는 단일상품인 Transportation과 그 변수들을 예시한다.

<표 5> Individual Component Transportation to composite Traveling Products

Individual Components	Variables (Preference)	Values
Transportation	Rental (0.4)	Car / Van
	Bus (0.6)	Tour / public
	Taxi (0.2)	Car / Van
	Train (0.6)	Train
	Subway (0.8)	Subway

<표 6>은 단일상품인 City-guide와 그 변수들을 예시한다.

<표 6> Individual Component City-guide to composite Traveling Products

Individual Components	Variables (Preference)	Values
City-Guide	Museum	In
	Castle	Out
	Tower	In
	Square	Out
	Stadium	Out

Step 2 : Components Modification

Step 2-1 : Value Change

추출된 단일상품들의 변수의 값이 고객 요구사항과 같지 않으면 요구해당 값으로 대체한다.

Global variables {*Member, Date, Place, Total-Budget*}

Global variables values {"Adult" "10-Oct-05, 20-Oct-05" "London, Hong Kong, Seoul" "<\$10,000"}

Components 1 : Airline

Must variables {*air-departure-date, air-arrival-date*}

Must variables values {"10-Oct-05", "20-Oct-05"}

Optional variables {*air-dinner, air-class, air-seat-number, air-route, air-price*}

Required optional variables values {"Special" "Business" "A15" "via Hong Kong" "<\$600"}

Selected optional variables values {"Regular" "Economy" null "via Tokyo" "<\$500"}

Modified optional variables values {"*Special*" "*Business*" "A15" "via Hong Kong" "<\$500"}

Components 2 : Accommodation

Must variables {*acc-departure-date, acc-arrival-date*}

Must variable values {"10-Oct-05" "20-Oct-05"}

Optional variables {*acc-#-of-room, acc-room-type, acc-price, acc-place*}

Required optional variables values {"1" "Double" "<\$150" "Seoul"}

Selected optional variables {"1" "Single" "<\$130" "Seoul"}

Modified optional variables {"1" "*Double*" "<\$130" "Seoul"}

Components 3 : Transportation

Must variable { }

Optional variables {*trans-rental, trans-bus, trans-taxi, trans-train, trans-subway*}

Required optional variables values {"Car" "City-Tour" null null "Subway"}

Selected optional variables values {null "City-Tour" null null "Subway"}

Modified optional variables values {"*Car*" "City-Tour" null null "Subway"}

Components 4 : City-Guide

Must variable { }

Optional variables {*city-museum, city-castle, city-tower, city-square, city-stadium*}

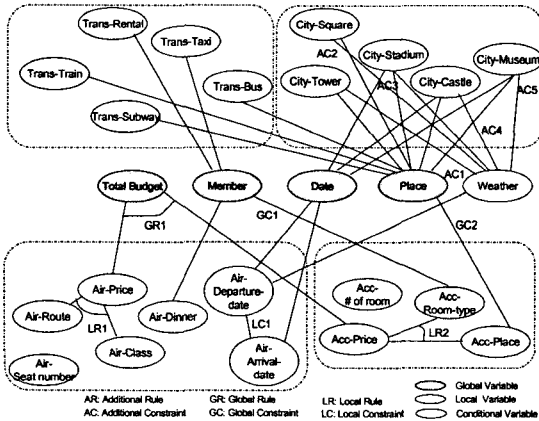
Required optional variable values {"In" "Out" "In" "Out" null}

Selected optional variables values {"In" null "*In*" "Out" null}

Modified optional variables values {"In" "*Out*" "*In*" "Out" null}

Step 2-2 : Consistency Check

[그림 3]은 여행 복합상품을 구성하는 변수들간의 연관관계와 제약조건 및 규칙들을 보여준다.



[그림 3] 변수들간의 제약적 연관 관계

다음 예의 acc-price와 air-price는 각각 Total-Budget과 연관이 있고 복합상품에서 동시에 Total-Budget에 영향을 주므로 다음과 같이 새로운 제약 조건이 생성되었다.

*IF (Airline IS 단일상품
 AND Air-price LE Total-budget)
 AND (Accommodation IS 단일상품
 AND Acc-Price LE Total-budget)
 THEN (Composite product IS 복합상품
 AND Air-price + acc-price LE Total-budget)*

<표 7>의 제약조건관 규칙들은 기존의 단일 상품 내에서의 변수들간의 연관관계를 이용하여 복합상품에서 새로이 구성된 제약조건과 규칙들간의 관계를 보여준다. 지역변수 내의 연관관계 규칙은 LR로 제약조건은 LC로 표현된다. 전역변수와의 연관관계의 제약조건은 GC로 규칙은 GR로 표현된다.

<표 7> Constraints and Rules

Constraints and Rules	
GC1 :	Constraints pairs of Member and acc-room-type {{{(Single) (Single)}} ((Double) (Double)) ((Family) (Family))}
GC2 :	Constraints pairs of Place and acc-place {{{(Seoul) (Seoul, Kyunggi)}}
...	
GC3 :	Total-budget ≤ air-price + acc-price
...	
LC1 :	air-departure-date < air-arrival-date
...	
LR1 :	Rule of air-route, air-class, and air-price
If (air-route is Tokyo or HongKong) and (air-class is Economy)	Then air-Price is \$500
If (air-route is Beijing or Manila) and (air-class is Economy)	Then air-Price is \$700
If (air-route is Tokyo or HongKong) and (air-class is Business)	Then air-Price is \$1000
If (air-route is Beijing or Manila) and (air-class is Business)	Then air-Price is \$1400
If (air-route is Tokyo or HongKong) and (air-class is First)	Then air-Price is \$1500
If (air-route is Beijing or Manila) and (air-class is First)	Then air-Price is \$2000
LR2 :	Rule of acc-place, acc-room-type and acc-price
If (acc-place is Seoul) and (acc-room-type is Single)	Then acc-Price is \$130
If (acc-place is Seoul) and (acc-room-type is Double)	Then acc-Price is \$150
If (acc-place is Seoul) and (acc-room-type is Family)	Then acc-Price is \$170
If (acc-place is Kyunggi) and (acc-room-type is Single)	Then acc-Price is \$70
If (acc-place is Kyunggi) and (acc-room-type is Double)	Then acc-Price is \$100
If (acc-place is Kyunggi) and (acc-room-type is Family)	Then acc-Price is \$120
...	
AC1 :	Constraints pairs of Weather and city-tower {{{(rainy) (in)}} ((sunny) (In, Out)) ((snowy) (Out))}
AC2 :	Constraints pairs of Weather and city-square {{{(rainy) (in)}} ((sunny) (In, Out)) ((snowy) (Out))}
AC3 :	Constraints pairs of Weather and city-stadium {{{(rainy) (in)}} ((sunny) (In, Out)) ((snowy) (Out))}
AC4 :	Constraints pairs of Weather and city-castle {{{(rainy) (in)}} ((sunny) (In, Out)) ((snowy) (Out))}
AC5 :	Constraints pairs of Weather and city-museum {{{(rainy) (in)}} ((sunny) (In, Out)) ((snowy) (Out))}

다음은 복합상품의 제약조건과 규칙들을 XRMML을 이용하여 표현하였다.

XRMML을 이용한 Constraints 표현은 다음과 같다.

```
<rule rid=1>
<RuleTitle>GC1 : Constraints pairs of Member and
acc-room-type</RuleTitle>
{{{(<IF><variable vid=1 item="Member"/><value
vid=1>Single </value></IF>)
(<THEN><variable vid=2 item="acc-room-type"/><value
vid=2> Single</value></THEN>)
((<IF><variable vid=1 item="Member"/><value
vid=1>Double
</value></IF>)
(<THEN><variable vid=2 item="acc-room-type"/><value
vid=2> Double</value></THEN>))
((<IF><variable vid=1 item="Member"/><value
vid=1>Family </value></IF>)
(<THEN><variable vid=2 item="acc-room-type"/><value
vid=2> Family</value></THEN>))}}
</rule>
```

```
<rule rid=2>
<RuleTitle >GC2 : Constraints pairs of place and
acc-place</RuleTitle>
{{{(<IF><variable vid=3 item="Place"/><value
vid=3>Seoul</value></IF>)
(<THEN><OR><variable vid=4 item="acc-place"/><value
vid=4> Seoul</value><<value vid=4>
Kyunggi</value></OR></THEN>}}
```

```
<rule rid=3>
<RuleTitle>GC3 : Rule of air-price, acc-price and
total-budget </RuleTitle>
If (<IF><LE><variable vid=5>Total-Budget</variable> <
<value vid=5>air-price +
acc-price</value></LE></IF>) Then
<THEN><variable vid=6
item="ServiceComposition"/><value
vid=6>Fail</value></THEN>
```

```
<rule rid=5>
<RuleTitle>LC1 : Rule of air-departure-date and
air-arrival-date </RuleTitle>
If ((<IF><LE><variable vid=9>air-departure-date
</variable> < <value vid=9>air-arrival-date
</value></LE></IF>)
<THEN><variable vid=10
item="ServiceComposition"/><value
```

```
vid=10>Fail</value></THEN>
</rule>
```

```
<rule rid=7>
<RuleTitle>AC1 : Constraints pairs of Weather and
city-tower </RuleTitle>
{{{(<IF><variable vid=14 item='Weather'/><value
vid=14>rainy</value></IF> (<THEN><variable
vid=15 item='city-tower'/><value
vid=15>in</value></THEN>)) ((<IF><variable
vid=16 item='Weather'/><value
vid=16>sunny</value>) (<THEN><OR><variable
vid=17 item=' city-tower'/><value
vid=17>In</value>, <value
vid=17>Out</value></OR><THEN>))
((<IF><variable vid=18 item='Weather'/><value
vid=14>snowy</value></IF>) (<THEN><variable
vid=18 item='city-tower'/><value
vid=18>Out</value></THEN>))}}
```

XRMML을 이용한 규칙 (Rules) 표현은 다음과 같다.

```
<rule rid=4>
<RuleTitle>LR1 : Rule of air-route, air-class, and
air-price </RuleTitle>
If (<IF><AND><OR><variable
vid=7>air-route</variable> is <value
vid=6>Tokyo</value> or <value
vid=7>HongKong</value></OR>) and (<variable
vid=8>air-class</variable> is <value
vid=8>Economy</value></AND></IF>)
Then <THEN><variable vid=8>air-Price</variable> is
<value vid=8>$500</value></THEN>
</rule>
```

```
<rule rid=6>
<RuleTitle>LR3 : Rule of acc-place, acc-room-type and
acc-price</RuleTitle>
If (<IF><AND><variable vid=11>acc-place</variable> is
<value vid=11>Seoul</value>) and (<variable
vid=12>acc-room-type</variable> is <value
vid=12>Single</value></AND></IF>)
Then <THEN><variable vid=13>acc-price</variable>
is<value vid=13>$130</value></THEN>
</rule>
```

제약조건과 규칙들에 의해 일치성을 확인하면 다음과 같다.

Components 1 : Airline

Optional variables {*air-dinner*, *air-class*, *air-seat-number*, *air-route*, *air-price*}

Modified optional variables values {"Meal" "Business" "A15" "via Hong Kong" "<\$500"}

Confliction variables {"Meal" "Business" "A15" "via Hong Kong" "<\$500"}

*air-class*가 "Business", *air-route*가 "HongKong", *air-price*가 "\$500"인 관계에서 충돌이 발생하였다. 즉 *air-class*가 "Business"이면서 "Hong Kong"을 경유하여 "\$500" 이내로 비행기표를 구입이 불가능하다. *Air-route* "Hongkong"은 "Tokyo"와 호환 가능하므로 *air-class* 또는 *air-price*의 값을 변경해야 한다. 이때 *air-price*는 전역변수인 *Total-budget*에 영향을 미치게 된다.

Components 2 : Accommodation

Optional variables {*acc-#of-room*, *acc-room-type*, *acc-price*, *acc-place*}

Modified optional variables values {"1" "Family" "<\$130" "Seoul"}

Confliction variables values {"1, "Family" "<\$130" "Seoul"}

*Acc-room-type*이 "Family"이고 *acc-price*가 "\$130"이하이면서 *acc-place*은 "Seoul"로 하여 숙박을 정하기 어렵다. 따라서 *acc-room-type*을 바꾸거나, *acc-price*을 늘리거나, *acc-place*를 변경할 수 있다. *acc-price*을 늘리면 전역변수인 *Total-budget*에 영향을 주며, *acc-place*를 변경하면 전역변수의 *Place*에 영향을 줄 수 있다. 또한 *Place*의 값의 변화는 다른 Components 3의 Transportation에, Components 4의 City-guide에 영향을 준다.

Components 3 : Transportation

Optional variables {*trans-rental*, *trans-bus*, *trans-taxi*, *trans-train*, *trans-subway*}

Modified optional variables {"Car" "City-Tour" null null "Subway"}

Confliction variables {}

Components 4 : City-guide

Optional variables {*city-museum*, *city-castle*, *city-tower*, *city-square*, *city-stadium*}

Modified optional variables values {"In" "Out" "In" "Out" null}

Confliction variables {}

Transportation와 City-guide에서는 충돌이 발생하지 않았다. 변수들간의 값의 일치성 검사는 지역변수의 값 변화에 따라 전역변수와의 관계와 각 component내에서 지역변수들 간의 값의 일치성 검사를 해야 한다.

Step 2-3 : Components or optional variable addition

새로운 단일상품 또는 변수가 복합상품에 추가될 때 복합상품의 변수들과의 연관관계에 따라 전역변수와 단일상품을 정한다.

다음의 변수 Weather의 결합 예는 다음과 같다. Weather는 단일상품인 City-guide와 Air-line에 영향을 주므로 전역변수로 추가 된다.

Weather {rain}

Components 1 : Airline

Must variables {*air-departure-date*, ...}

Components 4 : City-guide

Optional variables {*city-museum*, *city-castle*, *city-tower*, *city-square*, *city-stadium*}

Step 3 : re-Composition

Step 3-1 : Conflict Resolution

1) 지역변수와 전역변수간의 충돌해결

지역변수와 전역변수 간에 충돌이 생기면, 고객의 선호도에 따라 지역변수와, 전역변수 중에 바꿀 값을 결정한다. 고객의 요구사항의 변수우선순위에 의하면 전역변수가 지역변수의 상위 개념이며 전역변수는 다른 단일상품과도 연관되므로 변수들간의 연관도가 높다. 따라서 고객의 선호도가 같은 경우는 지역변수의 값을 변환한다.

- 2) 지역변수가 속한 단일상품내의 다른 지역변수와 충돌이 발생하면 지역변수들 중 선호도가 낮은 변수의 값을 바꾼다 [12].

Component 1 : Airline

Optional variables {*air-dinner*, *air-class*, *air-seat-number*, *air-route*, *air-price*}

Confliction variables {"Meal" "Business" "A15" "via Hong Kong" "<\$500"}

*air-class*와 *air-price*의 고객 선호도는 같지만, *air-price*가 전역변수인 *Total-budget*과 연관되므로 *air-class*의 값을 변경한다. *air-class*의 값을 "Business"에서 "Economy"로 변경하면 다음과 같다.

Modified variables values {"Special" "Economy" "A15" "via Hong Kong" "<\$500"}

Component 2 : Accommodation

Optional variables {*acc-#-of-room*, *acc-room-type*, *acc-price*, *acc-place*}

Confliction variables {"1" "Family" " <\$120" "Seoul"}

지역변수 *acc-room-type*, *acc-price*, *acc-place*는 모두 전역변수와 연관관계를 가진다. *acc-price*는 *Total-budget*에 영향을 주고, *acc-room-type*은 *Member*에 영향을 주며, *acc-place*는 *Place*에 영

향을 준다. *Place* "Seoul"에 대해 *acc-place*는 "Seoul", "Kyunggi"가 호환 가능하므로 *acc-place*의 값은 호환 가능한 값 범위 내의 "Seoul"로부터 "Kyunggi"로 바뀔 수 있다. *acc-place*가 "kyunggi"로 하고 *acc-room-type*이 "Family"가 되면 *acc-price*는 "\$120"이므로 \$130보다 적어야 하는 조건을 만족한다.

Modified variables values {"1" "Family" " <\$120" "Kyunggi"}

*acc-place*의 값의 변경은 호환가능한 값 범위 내에서 변경되었으므로 전역변수와 단일상품내의 지역변수들에게 영향을 주지 않는다. 예를 들어 *Place*의 값이 변경되면 *Transportation*이나 *City-guide*의 지역변수들에게 영향을 미치게 된다.

*Component 3 Transportation*과 *Component 4 City-guide*에서는 *Confliction*이 발생하지 않았다. 복합상품에 *Weather* 변수가 추가 될 때 전역변수인 *City-guide*에 영향을 준다.

Component 4 : City-guide

*Weather*의 value가 "rain"이면 *City-guide*의 외부활동에 해당하는 변수들이 영향을 받는다. 따라서 "Out"에 해당하는 값들이 null값으로 변환된다. 즉 *city-castle*과 *city-square*의 외부활동이 취소되었다.

Optional variables {*city-museum*, *city-castle*, *city-tower*, *city-square*, *city-stadium*}

Modified optional variables values {"In" "Out" "In" "Out" null}

Confliction variable {"In" "Out" "In" "Out" null}

Modified variables values {"In" null "In" null null}

4.3 실험 및 분석

본 실험에서는 기존의 시스템, 즉 웹 서비스와 상품묶음 프로세스를 지원하지 않는 시스템으로 유동소비자 개인이 필요로 하는 정보를 수집하여 복합상품을 구성하여야 하는 경우와 본 연구에서 제안된 시스템 간의 상품 검색 비용에 대해 비교 실험 하였다. 실험은 Netlogo 시뮬레이션 툴을 이용하였으며, 20명의 사용자가 100개의 사이트에서 4가지 여행 단일상품, 항공사, 호텔, 자동차대여 및 여행지정보를 복합상품화 하는 것을 대상으로 실험하였다. 연구가설은 H_0 : 상품묶음을 위한 웹 서비스 기반 탐색비용은 일반적 탐색의 비용과 같 다로 하였다. 227번의 실험을 통해 양 시스템의 탐색비용을 비교한 결과 기존시스템과 본 연구에서 제안 시스템 간의 탐색비용은 통계적 유의수준에서 차이가 있음을 확인하였다($t=24.47, p < 0.0001$). 즉 웹 서비스 기반 PBM이 일반적 탐색시스템보다 효율적임을 확인 할 수 있었다.

5. 결론 및 향후 연구방향

유비쿼터스 환경에서 유동 소비자가 언제 어디서나 원하는 서비스를 제공받을 수 있는 여행정보 서비스를 묶음상품 방법론과 구조적인 정보 처리 기능을 중심으로 설계하고 방법론을 제안하였다. 묶음 상품 방법론으로 Product Bundling and Modification 알고리즘을 제안하였고, 알고리즘의 핵심은 단일상품이 복합상품을 이루는 방법론으로 단일상품들간의 전역변수와 단일상품내의 지역변수들을 구분하고 상품구성에서 발생하는 변수들의 값들간의 상충을 해결하고자 하였다. 즉, 여행서비스 상품을 대상으로 효율적 복합상품 구

성을 위한 방법론을 제안한다. 또한 본 논문에서 복합상품을 구성하기 위해 공유되어야 할 데이터는 XML로, 복합상품 구성 시에 준수해야 할 규칙은 XRML로 표현 함으로써 이질적인 정보시스템이 이들 정보를 공유할 수 있는 기반을 제시하였다. 향후에는 시맨틱 웹 서비스 기반의 여행정보서비스 시스템 구축을 위해서 서비스 중심의 아키텍처와 이질적인 데이터들을 통합/활용하기 위한 온톨로지의 구축 및 통합방법에 대한 연구가 요구된다.

참고문헌

- [1] Adomavicius, G. and A. Tuzhilin, "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(2005), 734-749.
- [2] Antonius G. and F. V. Harman, "A Semantic Web Primer", The MIT Press, 2004.
- [3] Berners-Lee, T., J. Hendler, and O. Lassila, "The Semantic Web : A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities", *Scientific American*, (2001), 34-43.
- [4] Choueiry, B. Y. and G. Noubir, "On the Computation of Local Interchangeability in Discrete Constraint Satisfaction Problems", *in : Proc. AAAI-98*, (1998), 326-333.
- [5] Delgado, J. and R. Davidson, "Knowledge bases and user profiling in travel and hospitality recommendation systems", *in : Proc. The Enter 2002 Conference*, Springer Verlag, Wien, NY, (2002), 1-16.
- [6] Gershman, A., "Ubiquitous Commerce - Always On, Always Aware, Always Pro-

- active”, *IEEE International Symposium on Applications and the Internet (SAINT 2002)*, 2002.
- [7] Gilmore, J. H., and B. J. Pine, “The Four Faces of Mass Customization”, *Harvard Business Review*, (1997), 91-101.
- [8] Gorden, B. D., “Anytime/Anyplace Computing and the future of knowledge work,” *Communication of the ACM*, Vol.45, No.12 (2002).
- [9] Gordijn, J., “Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas”, Ph.D. diss., Dept. of Computer Science, Vrije Univ., 2002.
- [10] Gruber, T., “A translation approach to portable ontologies”, *Knowledge Acquisition*, Vol.5, No.2(1993), 199-220.
- [11] Lampel J. and H. Mintzberg, “Customizing Customization”, *Sloan Management Review*, (Fall 1996), 21-30.
- [12] Lee, H. J. and J. K. Lee, “An Effective Customization Procedure with Configurable Standard Models”, *Decision Support Systems*, Vol.41, No.1(2005), 262-278.
- [13] Lee, J. K. and M. Sohn, “eXtensible Rule Markup Language”, *Communications of the ACM*, Vol.46, No.5(2003), 59-64.
- [14] Lyytinene, K. and Y. Yoo, “Research Commentary : The Next Wave of Nomadic Computing”, *Information Systems Research*, Vol.13, No.4(2002), 377-388.
- [15] Ricci, F. and H. Werthner, “Case-based querying for travel planning recommendation”, *Information Technology and Tourism*, Vol.4, No.3(2002), 215-226.
- [16] Shin N., “Strategies for competitive advantage in electronic commerce”, *Journal of Electronic Commerce Research*, Vol.2, No.4(2001).
- [17] Weng S. S. and M. Liu, “Personalized product recommendation in E-Commerce”, *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE'04)*, (2004), 413-420.
- [18] Weiser, M., “Ubiquitous Computing”, *IEEE Computer*, 1993.
- [19] Werthner, H. and F. Ricci, “E-Commerce and Tourism”, *Communication of ACM*, Vol.47, No.12(2004), 101-105.
- [20] Xiao B., E. Aimeur and F. Fernandez, “PCFinder : An Intelligent Product Recommendation Agent for E-Commerce”, *IEEE International Conference on E-Commerce Technology (CEC'03)*, (2003), 181-188.
- [21] 김우주, 최대우, 진봉균, “에이전트 쇼핑몰을 위한 의미 웹 서비스 기반 지능형 상품 정보 검색 프레임”, ITRC Forum, 2004.
- [22] OWL-S Coalition. OWL-S 1.0 Release. At <http://www.daml.org/services/owl-s/1.0/>
- [23] The DAML Services Coalition, 2003, DAML-S (0.9 Draft Release. <http://www.daml.org/services/daml-s/0.9/>
- [24] W3C, Web services architecture, <http://www.w3.org/TR/ws-arch/>, Feb. 2004.

Abstract

Traveling Product Bundling on Web Service Composition in Ubiquitous Computing Environment

Hyun Jung Lee* · Mye M. Sohn**

In this research, we are suggesting intelligent information system for traveling which is focusing on product bundling and integration of information from various resources on ubiquitous computing environment. It is necessary for products structure to easily integrate according to customers' requirements because traveling product is integrated by various traveling resources like airline, hotel reservation, and so on. To guarantee of traveler's mobility in ubiquitous computing environment, we need product bundling and modification process to configure products and semantic web service which supports ontology based traveling information system to support immediate integrating of traveling information from various resources. In this research, we offer a product bundling and integration of information. It is based on the semantic web service, with several components (single products) to reconfigure a bundle of traveling products.

Key words : Traveling system, Product bundling, Ubiquitous computing, Product structure, Ontology, Semantic web service

* Sungkyunn Institute of Management Research, Sungkyunkwan University

** Department of Systems Management Engineering, Sungkyunkwan University