

의미웹에서 한정도메인 제약식을 이용한 지능형 쇼핑에이전트 : CD 쇼핑몰의 경우를 중심으로

김학진
연세대 경영대학 경영학과
(hakjin@yonsei.ac.kr)

이명진
연세대학교 공과대학 정보산업공학과
(xml@yonsei.ac.kr)

.....

인터넷을 통한 온라인 구매에 소비자들은 현 탐색엔진 및 웹 구조의 한계와 의사결정 도구의 부족으로 많은 어려움을 겪는다. 이 논문은 인터넷 쇼핑의 상황에서 소비자가 결정해야 하는 의사결정 문제를 상정하고 지능형 에이전트 구축을 통하여 그 의사결정 과정을 돕는 의사결정의 틀을 제시한다. 이 에이전트는 의미 웹 환경에서 한정도메인 제약식 프로그래밍을 추론엔진으로 삼아 의사결정을 돕는다. 이를 통해 의미웹과 제약식 프로그래밍의 두 기술의 결합이 인터넷 쇼핑 시 소비자가 겪게 되는 어려움을 어떻게 해결하는지를 제시한다.

.....

논문접수일 : 2006년 06월 게재확정일 : 2006년 12월 교신저자 : 김학진

1. 서론

21세기 전자 상거래 시대에 들어서면서 많은 소비자들이 인터넷 온라인 상점에서 제품을 구입하는 경향을 보이고 있다. 이는 현대 사회가 개인에게 한정된 시간 안에 많은 복잡한 일들을 할 것을 요구하는 상황으로 말미암아 소비자가 오프라인 상점을 찾아 다니면 구매할 시간이 부족하게 되었다는 것이 한 이유가 될 수 있다. 그리고 인터넷을 통해 구매를 할 때 탐색 엔진을 통해 여러 쇼핑몰의 제품가격 및 판매 조건 등을 비교하여 최적의 구매 의사 결정을 할 수 있다는 장점이 있기 때문이기도 하다. 그러나 이런 이유를 자세히 살펴보면 한 소비자가 실제로 인터넷의 탐색 엔진을 통해서 여러 쇼핑몰의 제품을 비교하고 구매하는 것은 생각하는 것만큼 단순하게 이루어지지 않

는 것을 알 수 있다. 현 탐색 엔진의 한계와 웹의 구조적인 문제 그리고 소비자가 원하는 최적의 의사결정을 도와주는 도구의 부재로 인터넷을 통하여 구매를 한다는 것은 쉬운 일이 아니다. 이 논문은 바로 인터넷 쇼핑몰을 통해 제품을 구매할 때 최적의 의사결정을 위해 필요한 틀을 에이전트의 구현을 통해 보여준다. 즉 한정도메인 제약식 프로그래밍을 이용하여 의미 웹 기반에서 최적의 구매 의사결정을 하는 에이전트를 구현하고자 한다. 이는 의사결정을 위해 먼저 의미 웹의 구조와 기능을 이용하여 필요한 데이터를 추출하고 이를 이용해 의사결정 문제를 한정도메인 제약식 프로그래밍으로 모형화하며 한정도메인 제약식 프로그래밍 솔버를 이용하여 최적의 해를 도출해 소비자에게 제시한다.

관련된 연구로서 다음 절에서 설명된 SWRL기

반의 제약식 언어인 CIF/SWRL이 있다(Gray et al., 2001). 이 논문에서 저자의 기본적인 접근방식은 의미웹의 개념들 특히 SWRL의 언어적인 추론 능력을 확장하기 위해 제약식 언어를 제시하는 것이다. 이는 기존의 제약식 언어가 SWRL과 마찬가지로 룰 기반의 논리프로그래밍 언어 위에서 구현된 것과 동일한 확장이라 볼 수 있다. 그리고 궁극적으로 제약식 술어를 이용해 의사결정 문제를 풀고자 하는데 있어서 본 논문과 같은 목적의식을 공유한다. 하지만 제약식 언어가 인위적인 문제가 아닌 실제 문제를 해결하는 데에는 여러 가지 해결해야 할 점들이 있다. 특히 오랫동안 실제 문제 해결에 많은 연구가 이루어진 OR의 시각에서 볼 때 문제의 표현은 항상 문제의 해법과 관련성을 가진다. 즉 문제 표현 언어는 문제의 해법에 의해 제한된다는 것이다. 예를 들어 선형계획법, 정수계획법 등의 단어는 문제 표현언어에 대한 설명인 동시에 모형의 해법에 대한 설명이다. 이 사실이 시사해 주는 것은 결국은 실제 문제의 확인과 이를 풀 수 있는 알고리즘적인 한계 그리고 이 안에서 문제를 표현하는 표현 언어의 구축으로 아래로부터의 접근 역시 중요하다는 것이다. 현재까지 구축된 제약식 술어들이 의미웹의 표현 언어와 유연하게 연결될 수 있는 언어적인 강한 표현 능력에도 불구하고 아직까지 실제 문제를 다루는데 있어서 OR 기법들에 비해 한계(예로써 이산형과 연속형 모형을 풀 수 있는 술어의 부재 등)가 있다는 것은 이를 확인해 준다. 본 연구는 의미 웹에서 추출될 수 있는 정보를 이용해 실제로 발생하는 의사결정 문제를 제약식 프로그래밍 언어로 설정하고 이를 제약식 술어로 푸는 일련의 의사결정 과정의 틀을 제시함으로써 표현언어 구축의 아래로부터의 접근법의 예를 보여주하고자 한다. 그리고 아직 의미웹과 제약식 언어의 결합이 해법적인 측면

에서 한계가 있음에도 여전히 실제 문제를 해결하는 유용한 틀임을 보여주고자 한다. 그럼으로써 CIF/SWRL의 기반 위에 이루어지지 않는 않았지만 CIF의 접근법의 유용성을 증명하는 한 예가 될 수 있다.

이 논문은 여섯 개의 절로 구성된다. 1절은 서론이고 2절은 관련된 연구주제에 대한 문헌연구가 이어진다. 3절에서는 이 논문에서 해결하고자 하는 문제를 정의하고 문제 해결의 각 단계마다 의미웹과 한정도메인 제약식 프로그래밍이 문제해결에 어떤 도움을 주는지를 서술한다. 4절에서 의미 웹과 한정도메인 제약식 프로그래밍을 이용한 에이전트의 전반적인 구조를 설명하고 5절에서는 문제를 위해 의미웹의 구조가 어떻게 형성되고 이로부터 어떻게 정보의 추출이 어떻게 일어나는지를 설명한다. 그리고 6절에서 의미웹에서 추출된 정보로부터 의사결정에 필요한 제약식 프로그래밍 모형과 해를 제시한다. 마지막 절에 결론 및 한계와 앞으로의 연구 방향이 그 뒤를 잇는다.

2. 관련 연구들

온톨로지(Ontology)는 실 세계에 존재하는 실체에 대한 개념화된 명세로써, 특정 언어로 정의된 개념들에 대한 기술과 개념들 사이의 관계를 나타낸다. Tom Gruber에 따르면 “온톨로지는 개념화의 명확한 명세이다(An ontology is an explicit specification of a conceptualization)”라고 정의하였다(Gruber, 1993). 온톨로지의 표현 언어로서 RDF(Miller, 1998)는 구조화된 메타 데이터를 생성하고 활용하기 위한 지능형 웹의 기반 기술이다. OWL(Antoniou and van Harmelen, 2003)은 웹에 존재하는 자원을 기술하기 위한 시맨틱 마크업 언

어로서, RDF를 기반으로 확장된 온톨로지 표현 언어이다. 이 외에 온톨로지 추론을 위한 룰 표현 언어인 SWRL(Horrocks et. al., 2004)과 온톨로지에 대한 질의를 위한 SPARQL(Prud'hommeaux and Seaborne, E., 2005) 등 지능형 웹의 언어에 대한 다양한 연구가 현재까지 이루어지고 있다.

조직과 기업의 인적자원의 개별적인 지식들을 체계화하고 공유함으로써 경쟁력을 향상시키는 지식경영이 중요한 성공요인으로 인식되고 있는데, 온톨로지 기반의 지식 관리 프레임워크에 관한 연구가 활발히 진행되고 있다. OTKM(Sure et. al., 2003)은 지식 프로세스와 지식 메타 프로세스 중심으로 타당성 조사, 진입, 정제, 평가, 응용 및 개선의 단계를 통해 지식의 공유와 재사용이 가능한 효율적인 온톨로지 시스템을 구현할 수 있는 방법론을 제시하고 있다. 또한 다양한 분야에서 온톨로지에 대한 필요성이 높아짐에 따라 의료 분야에서의 대용량 온톨로지 개발(Hahn and Schulz, 2003)뿐만 아니라 많은 온톨로지들의 개발이 이루어지고 있다. 이에 발맞추어 온톨로지 간의 병합과 맵핑 또한 중요한 연구 분야로 자리잡게 되었다. (Mitra and Wiederhold, 2003)의 연구에서는 온톨로지의 상호 교환 및 통합을 위해 Articulation 방법을 이용해서 단어의 동의어 및 상하위 관계와 같은 의미적인 관계를 정의함으로써 하나 이상의 온톨로지 병합에 대한 문제를 다루고 있다. 이 외에도 온톨로지에 관한 연구는 온톨로지 학습(Maedche and Staab, 2003)에 관한 연구뿐만 아니라 Protégé(Noy et. al., 2001)와 같은 온톨로지 저작 툴 및 OntoBroker(Decker et. al., 1999)와 같은 온톨로지 서버, 그리고 온톨로지를 이용한 응용 어플리케이션까지 그 연구 범위가 실로 다양하다.

인터넷의 발달로 온라인(online)에서의 구매 활동이 더 활발해지고 있다. 하지만 현재까지의 온라

인 쇼핑은 구매자가 직접 각 사이트를 방문하면서 상품을 검색하고 비교하는 수준에 머물고 있는 실정이다. 자동화된 쇼핑을 위해 쇼핑 에이전트(Greenwald and Kephart, 1999)를 이용하여 해당 상품을 판매하고 있는 다양한 쇼핑몰로부터 실시간으로 가격 정보를 추출하고 가격을 비교하여 그 결과를 알려줌으로써 구매자에게 편의성을 제공(Pack, 2001)해주고 있기는 하지만 최적의 선택을 하기 위한 구매자의 노력은 전과 다르지 않다. 최근에 들어서면서 보다 진보되고 지능화된 쇼핑 에이전트에 대한 연구가 활발해지고 있으며, 연구 중 하나로 IntelliShopper(Menczer et. al., 2002)에 대한 연구에서는 멀티 에이전트(Multi-Agents) 시스템을 이용하여 사용자의 행동에 따라 학습함으로써 보다 지능화되고 개인화된 쇼핑 에이전트의 아키텍처와 컴포넌트를 소개하고 있다. 또한 (Wen-Shan, 2005)의 연구에서는 지능형 에이전트와 지능형 쇼핑 에이전트를 소개하고 현재의 문제점을 분석하면서, 의사 결정을 위한 지식 기반(Knowledge-based)의 쇼핑 에이전트에 대해 다루고 있다.

최근에 들어서 의미 웹에서의 제약식에 대한 개념이 도입되어 의미 웹의 실제 활용 가능성을 높이고 있다. (Berners-Lee, 1999)가 웹 정보를 컴퓨터를 이용하여 가공 처리할 수 있도록 의미를 주는 의미 웹 개념이 제안된 이후로 연구는 주로 정보를 웹에 어떻게 저장하고 표현하는가에 초점이 맞추어졌고 이 때문에 RDF, XML Schema, Owl 등이 제안되고 실현되었다. 그 이후 현재에 이르러 그 비전은 자연스럽게 웹에 저장, 표현된 데이터를 어떻게 이용하는가에 관심을 갖기 시작하였고, 이를 위해 논리, 규칙, 제약식 등의 표현을 이용하여 데이터를 추출하고 이로부터 저장된 웹 데이터를 이용하여 필요한 응용문제들에 표현하는 문제에 관심을 기울이게 되었다. 특히 제약식은 개체와 개

체 간의 관계를 보는 새로운 패러다임으로서 문제를 표현하는 데에도 강점을 지니고 있는 것으로 알려지면서 최근 관심이 부각되어 SWRL의 기반 위에 제약식을 표현하는 CIF/SWRL을 제안하게 되었다(Gray et. al. 2001). 여기서 저자는 제약식 언어인 Colan (Bassiliades and Gray, 1994)과 이 위에서 작동하는 데이터 모형을 제시하여 RDF 스키마를 통해 제약식 데이터 모형을 사용할 수 있는 인터페이스를 제시하고 있다. 또한 (Preece et. al., 2006)은 의미웹에서 소프트 제약조건식의 다루는 방법을 다루고 있는데, 소프트 제약조건식이란 응용문제에서 엄격하게 충족되어야 하는 제약식이 아니라 충족시키는 정도를 감안하여 모든 제약식을 충족시키는 해가 없을 경우 중요하지 않은 제약식을 제거함으로써 좀더 약한 문제의 표현에서 만족하는 해를 찾을 수 있도록 제약식에 강제성의 정도가 부여 된 제약식으로 볼 수 있다. 위 논문은 현재 진행되는 SWRL 위에서 이루어지고 있는 제약식의 표현 상에 소프트 제약식을 어떻게 구현할 수 있는지에 대한 방안을 모색하는 논문이다.

3. 문제의 정의

이 절에서는 소비자가 인터넷에서 원하는 상품을 어떻게 구매하고 이때의 의사결정에 의미웹과 한정도메인 제약식 프로그래밍을 이용한 에이전트가 어떻게 원하는 의사결정 과정을 도울 수 있는지를 설명한다.

소비자가 인터넷을 통해서 구매하는 작업은 먼저 탐색을 통해서 일어난다. 예를 들어, 원하는 CD를 구매하고자 하면 탐색엔진을 이용하여 원하는 CD의 앨범 이름이나 그 앨범에 포함되어있는 노래 혹은 가수 이름 등의 그 CD에 대한 키워드를

탐색엔진에 입력하여 관련된 정보를 찾아낸다. 대부분의 경우 현재의 탐색엔진을 통해서 얻어진 결과들은 소비자가 검토하기에 너무 많은 정보를 주기 때문에 어느 사이트에서 원하는 CD를 구매할 수 있는지를 알려주기에 혼란을 준다. 특히 소비자는 검색된 결과를 바탕으로 각 쇼핑몰 사이트를 돌아다니면 그 쇼핑몰이 정말로 원하는 제품을 가지고 있는지를 확인해야 한다. 이는 일정 부분 현재의 탐색 방법이 단지 키워드에 바탕을 두고 있을 뿐 탐색 엔진 자체는 결과의 의미를 파악하지 못하기 때문이다. 이 해결의 한가지 방법으로 의미 웹 기술을 적용함으로써 정보 추출의 정확성을 기할 수 있다. 즉 OWL을 통한 온톨로지의 구현이 그것이다.

하지만 이것만으로 소비자의 구매에서의 모든 문제가 해결되는 것은 아니다. 일반적으로 소비자는 각 쇼핑몰에서 자신이 원하는 상품을 발견했을 경우도 그 물건을 구입하기 위한 조건들을 살펴보고 그 정보를 모아야 한다. 이런 조건으로는 쇼핑몰의 가격정책, 할인정책, 배송비의 처리 문제, 반환 정책 등 고려해야 할 많은 판매정책이 쇼핑몰마다 다 다르게 존재하므로 이러한 정보를 일일이 확인하고 모으는 것은 쉬운 일이 아니다. 의미웹의 기술은 이러한 조건들이 각 쇼핑몰의 온톨로지에 SWRL의 형태로 구축되어 있을 때 추출하여 판매정책을 확인하는 것을 용이하게 한다.

마지막으로 소비자가 원하는 상품을 가진 쇼핑몰을 확인하고 각 쇼핑몰의 판매정책을 확인하였을 때 소비자의 입장에서 최적의 상품과 쇼핑몰의 조합을 찾는 문제에 부딪힌다. 이 연구는 한정도메인 제약식 프로그램을 이용하여 소비자가 원하는 목표를 최적으로 만족하는 상품과 쇼핑몰 조합을 찾아냄으로써 소비자가 인터넷 쇼핑 시 부딪히는 문제를 해결하고자 한다.

제약식(constraint)은 많은 의사결정 문제에서 문제를 설명하는 도구로서 또 문제를 해결하는 도구로서 사용되어 왔다. 예를 들어 해당하는 숫자의 조합에 의해서만 열리는 자물쇠 문제(a bicycle number lock problem)에서 만약 숫자 조합 중 한 숫자를 기억하지 못한다고 할 때, 몇 가지 부분적인 정보가 주어지면 그 숫자를 찾아내는데 도움이 된다. 예로써 그 숫자가 1이 아니고 홀수이면서 소수가 아니라면 이런 조건을 만족하는 수는 9밖에 없다(Fruehwirth and Abdennadher, 2003). 의사결정의 문제는 이와 같이 해를 알지 못할 때 해의 성질을 설명하는 규칙들 즉 제약식으로 표현이 되었고 표현된 부분 정보를 이용하여 일정한 연산(calculus)를 통해 가능한 해를 찾아낸다. 일반적으로 제약식 프로그래밍의 솔버는 논리 프로그래밍의 기반 위에 구현되었는데 이는 논리 프로그래밍이 대입, 변수의 이름 변경, 두 개체의 일치화, 비확정성에 의한 탐색 등 제약식의 구현에 필요한 여러 개념들의 틀을 제공하기 때문이다. 논리 프로그래밍의 기반 위에 문제의 도메인이 갖게 되는 규칙들을 이용한 제약식 솔버를 통해 제약식의 부분 정보로부터 문제상황의 구체적인 해를 도출해 내게 된다. 이 논문에서 관심을 갖는 것은 앞에서 설명한 인터넷 구매 문제로 이 문제는 문제의 속성상 제약식 프로그래밍의 한정도메인(finite domain)을 이용해서 문제 설정이 가능하고 또 해당 솔버를 이용하여 풀 수 있다.

한정도메인 제약식 프로그래밍 솔버가 문제를 푸는 과정은 다음과 같다. 먼저 문제의 모형이 주어지면 솔버는 모형을 이용해 제약식 네트워크를 구현한다. 제약식 네트워크는 제약식과 제약식에 의해 값의 제한을 받는 변수들과의 관계를 나타낸다. 초기에 각 변수는 해당하는 초기 도메인을 가지게

되고, 그 도메인에 속한 각 값이 변수가 가질 수 있는 값의 대안이 된다. 솔버는 제약식 네트워크 위의 변수와 제약식들의 관계를 통해서 각 변수들이 주어진 제약식 조건들을 위배하지 않는 값을 갖도록 그 일관성을 유지한다. 그 일관성을 유지하기 위해 일반적으로 지역 확산법(propagation)을 사용하는데 이는 제약식의 형태에 따라 그리고 확산의 정도에 따라 다양한 형태의 기법들이 사용된다. 일반적으로 다루게 되는 문제들은 NP-hard의 경우가 많으므로 시간-메모리의 상충관계에 의해 완전한 일관성을 이루는데 많은 비용이 들어간다. 따라서 확산법은 보통 지역 확산법에 국한되며 전체적인 일관성을 이루기 위해 전역적으로는 탐색에 의존한다. 즉 변수의 값을 고정함으로써 변수의 도메인에 변경을 주고 이를 통해 확산법을 사용해 제약식에 위배되는 다른 변수들의 대안 값들을 제거해 나간다. 변수값의 고정은 탐색 가지치기와 같이 재귀적으로 일어나고 문제에서 유한의 제약식, 유한의 변수와 유한의 도메인을 가정하므로 결국 탐색 가지의 깊이는 유한할 수 밖에 없다. 이와 같이 전역적인 탐색과 지역적인 확산법으로 모든 제약식을 만족하는 변수의 값들만을 유지하게 된다. 이 논문의 문제처럼 최적의 해를 구하는 경우 솔버는 해를 평가하는 제약식을 추가하고 평가값에 해당하는 변수의 상한이나 하한 값을 탐색 시에 추가적으로 갱신해 줌으로써 최적의 값을 주는 해를 찾게 된다. 이 논문에서 한정도메인 제약식 프로그래밍 솔버로 GNU Prolog(gprolog)를 사용한다(Codognet and Diaz 1996; Diaz and Codognet 2001).

이 논문은 구매의 의도로부터 시작하여 구매를 위한 탐색, 정보의 추출, 그리고 구매의 결정까지 소비자의 의사결정과정을 자동적으로 도와주는 에이전트를 구축할 수 있음을 예를 통해 보여준다.

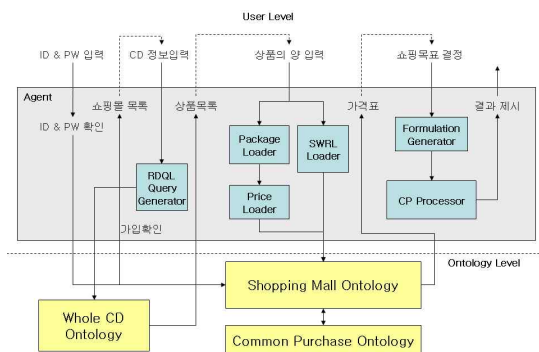
4. 지능형 에이전트의 전반적인 구조

[그림 1]은 에이전트의 문제 해결 과정을 보여 주고 있다. 그 과정을 설명하면 고객은 먼저 자신의 계정을 통해 에이전트에 접근한다. 이때 에이전트는 계정의 아이디와 암호를 확인하고 각 쇼핑몰의 온톨로지에 접근하여 고객이 거래를 위해 등록된 쇼핑몰의 정보를 리스트를 통해 제시한다. 에이전트의 활동은 고객이 필요한 상품의 정보를 입력함으로써 시작된다. 예로 고객이 구매를 원하는 CD의 타이틀이나 포함되어 있는 노래의 제목, 혹은 가수의 이름 등을 입력하면 에이전트는 쿼리 생성기에서 RDQL을 이용한 쿼리를 만들고 모든 CD의 정보를 저장 보관하고 있는 전체 CD 온톨로지(Whole CD Ontology)에 문의를 하여 주어진 정보에 따라 관련된 CD의 정보를 추출한다. 에이전트는 추출된 정보를 표의 형태로 정리하여 고객에게 보여주고 고객의 입력을 기다린다. 고객은 발견된 CD가 자신이 원하는 상품인지를 확인하고 이를 카트에 모은 후 쇼핑을 계속하거나 카트로 이동하여 각 상품의 원하는 양을 입력한다.

다음 단계로 에이전트는 각 쇼핑몰 온톨로지(Shopping Mall Ontology)에서 주어진 CD의 재고가

있는지를 확인하고 쇼핑몰의 비즈니스 원칙들에 대한 정보를 추출한다. 그리고 가격표(Price Table)을 제시함으로써 각 상품에 대한 가격, 배송비 그리고 할인 등의 정보를 모아 고객에게 알려준다. 이때 패키지로더(Package Loader)는 쇼핑몰에서 고객이 원하는 상품이 단일 상품인지 아니면 패키지에 포함이 되어있는지를 쇼핑몰 온톨로지에 확인해 보고 가격표에 그 정보를 기록한다. 또한 각 상품 별로 쇼핑몰 온톨로지서 추출한 가격 정보를 가격로더(Price Loader)를 통해 가격표에 제시한다. SWRL 로더는 쇼핑몰 온톨로지에 연결된 일반 구매 온톨로지에 SWRL 문서로 표현되어 있는 비즈니스 룰로부터 배송비와 할인에 대한 정보를 추출하여 이 정보를 가격표에 담아준다. 이렇게 해서 얻어진 가격표를 통해 에이전트는 고객에게 어떤 쇼핑몰에 고객이 원하는 상품 중 어떤 상품이 얼마만큼 있는지 그리고 각 상품의 판매 정책이 어떻게 되는지를 요약하여 제공한다.

마지막 단계로서 고객은 총비용을 최소화할지, 배송비를 최소화할지 혹은 포인트를 최대로 가져갈지 등 구매 의사결정의 목표를 설정하여 에이전트가 그 기준에 맞추어 최적화된 조합을 찾도록 그 기준을 제시하면 에이전트는 해당되는 목표에 적합한 의사결정 CP 모델을 만들고 CP 프로세서에서 이를 풀어 최적의 조합을 찾아준다.



[그림 1] 지능형 에이전트의 문제해결 과정도

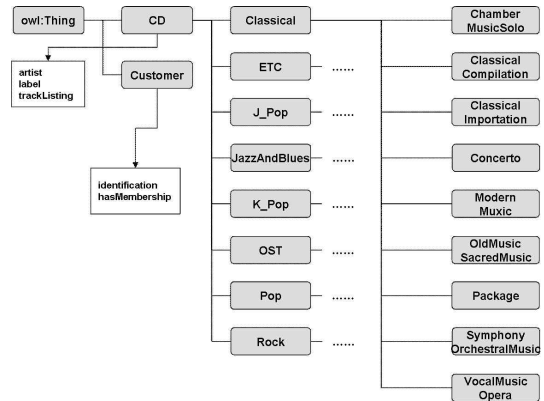
5. 의미웹 환경에서의 정보 추출

5.1 OWL을 이용한 온톨로지의 구현

전체 CD 온톨로지와 쇼핑몰 온톨로지는 실제 쇼핑몰의 카테고리에 따라 설정한다. 그 구현은 프 로테제 툴을 이용하여 OWL로 구현되었다. 전체

CD 온톨로지는 전체 쇼핑물이 가지고 있는 모든 CD에 대한 정보를 보관한다. 따라서 해당 CD에 대한 정보는 온톨로지에 대한 쿼리를 통해서 얻을 수 있다. 고객이 에이전트를 통해 앨범의 타이틀, 수록된 노래의 이름 그리고 가수의 이름 등을 탐색할 때 에이전트는 RDQL 같은 쿼리 언어를 통해 전체 CD 온톨로지에 질의함으로써 원하는 CD의 정보를 얻어내 고객에게 보여준다. 다음의 [그림 2]는 전체 CD 온톨로지가 어떤 식으로 구조화되어 있는지를 보여주고 있다. [그림 3]은 실제 XML 코드로 온톨로지가 구현되어 있는 양상을 보여준다.

이 연구에서 에이전트를 구현하기 위해 6개 쇼핑물의 정보를 사용하여 6개 쇼핑물 온톨로지를 구현하였다. 전체 CD 온톨로지와는 달리 각 쇼핑물 온톨로지는 모든 CD 정보를 수용하고 있는 것이 아니라 각 쇼핑물의 재고 상황에 따라 일정 부분의 CD 정보만을 가지고 있다. 또한 쇼핑물이 회원제로 운영될 경우 회원의 정보 또한 보관되어



[그림 2] Whole CD Ontology의 클래스 구조

고객이 에이전트를 통해 쇼핑물에 접속할 때 고객이 해당 쇼핑물의 회원임을 에이전트에게 확인해주는 기능을 담당하게 된다. 이외에도 고객의 구매에 필요한 판매 정책과 같은 주요 정보를 보관하고 있다. 일반적인 판매 정책으로는, 배송비의 문제, 상품이 반환되었을 때 어떻게 처리할 지의 문

```
<owl:Ontology rdf:about=""/>
<owl:Class rdf:ID="ForeignFilm">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="OST"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="KoreanFilm">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OST"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="JNBCompilation">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="JazzAndBlues"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="HardROckHeavyMetal">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Rock"/>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf="Group">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="KPOP"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf="Woman">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#KPOP"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf="WorldMusic">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="POP"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf="Trot">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#KPOP"/>
  </rdfs:subClassOf>
</owl:Class>
```

[그림 3] 전체 CD 온톨로지의 XML code

제, 할인 정책 및 포인트나 마일리지 등의 마케팅 전략에 따른 세부적인 정책들로 이루어진다. 이 쇼핑몰 온톨로지에서 재고로 존재하는 상품에 대한 정보는 전체 CD 온톨로지를 구현하는 방식과 동일하게 OWL을 이용하여 XML 코드로 구현된다. 하지만 판매정책과 같은 비즈니스 룰 등의 구현에 OWL을 이용할 수는 없다. 이는 기본적으로 OWL이 객체를 표현하기 위한 언어이지 객체간의 관계를 표현하기 위한 언어가 아니기 때문이다. 대신 RuleML에 따라 제안된 룰 표현언어인 SWRL을 이용하여 표현할 수 있다.

5.2 SWRL을 이용한 비즈니스 룰 표현

여러 가지 비즈니스 룰 중에 현재 중요하게 여겨지는 것은 전체 구매액에 대한 배송비 정책과 할인 정책 그리고 포인트나 마일리지 제도 등이 있다. 여기서는 그 중 하나인 배송비와 할인의 문

제를 예로서 다룬다. 포인트 마일리지 제도는 현재 해당 쇼핑몰 뿐만이 아니라 외부의 협력관계에 있는 회사로부터의 매출에서도 쇼핑몰의 포인트에 연계되기 때문에 여러 가지 고려되어야 할 사항 등이 복잡하다. 따라서 이 논문은 상대적으로 간단한 배송비와 할인 문제만을 고려 대상으로 삼아 고객의 구매 의사결정 과정과 제약식 프로그래밍 기법과의 연계를 살펴본다.

쇼핑몰의 비즈니스 룰을 표현하는데 중요한 두 가지의 사항이 있다. 하나는 그 표현이 누구에게라도 이해될 수 있도록 표준화되어 있어야 한다는 것이다. 쇼핑몰마다 그 표현 언어의 차이로 인하여 서로에게 이해되기 힘들면 안 되기 때문이다. 이 논문에서 제기하는 에이전트는 비즈니스 룰을 쿼리를 통하여 각 쇼핑몰 온톨로지에 문의한다. 이때 각 쇼핑몰의 룰 표현 언어가 다르면 고객이 원하는 상품과 쇼핑몰의 정보를 고객에게 제공하려는

| Product Price Range | Delivery Rates |
|-------------------------------|----------------|
| Greater than or equal ₩20,000 | ₩ 0 |
| less than ₩20,000 | ₩ 1,000 |

<CDPark>

[그림 4] CD Park 쇼핑몰의 배송비 정책과 SWRL 표현

에이전트의 목표가 달성되기 어렵다. 둘째 사항은 비즈니스 룰을 이미 알려진 정보와 연계하여 새로운 정보를 추출하기 쉬운 형태로 표현되어야 한다는 것이다. 따라서 만약 SWRL을 비즈니스 룰의 표현언어로 하고 그 표현이 인터넷 상에 위치하여 어느 누구라도 정보에 접근할 수 있다면 첫 번째 목표는 쉽게 달성된다. 또한 SWRL은 OWL과 연동될 수 있는 룰 표현 언어로 룰 기반의 추론에 편리한 강점이 있기 때문에 두 번째 사항도 해소될 수 있다.

현재 대부분의 쇼핑몰은 일정한 구매액 이상이 될 경우 배송비를 면제하도록 하는 정책을 취하고 있다. 좀더 일반적인 경우는 구간별 선형의 형태이지만 아직 이런 형태의 비용함수를 채택하지 않으므로 앞의 경우를 이 논문의 예로 상정한다. 그림 4는 쇼핑몰 CD Park의 배송비 정책을 보여주고 있고 이것이 프로테제 톨을 통해 SWRL으로 표현이 되고 있는 것을 보여주고 있다. [그림 5]는 그림 4에서 SWRL으로 표현되었던 배송비 정책을 자세히 보여주고 있다. 룰의 전제에 나타나는 첫째 술부는 “?mall” 변수가 CP 온톨로지의 “shoppingmall” 클래스에 속함을 보여주고 있다. 이때 CP는 [그림 1]의 일반구매 온톨로지(Common Purchase Ontology)를 나타내는 말로 일반 구매 활동을 정의하고 있으며 쇼핑몰 온톨로지 정보로 전달되어 SWRL의 단어(atom)들을 정의하게 된다. 그리고 술부 단어 “shoppingmall”은 변수 “?mall”이 쇼핑몰임을 나타낸다. 두 번째 술부의 “purchaseAmount”는 쇼

핑몰 “?mall”로부터 구매된 액수가 “?amount”이라는 사실을 말하고 셋째의 술부는 구매액이 20000보다 작다는 것을 의미한다. 따라서 이 세계의 술부가 다 만족이 되면 룰의 결과에 나타난 쇼핑몰 “?mall”의 배송비 1000이 부과됨을 나타낸다.

총 구매액에 따른 할인 역시 동일한 방식으로 나타난다. SWRL로 표현 시 필요한 각 단어는 일반구매 온톨로지에 정의되어 있고 쇼핑몰 온톨로지로 전달되어 사용된다. 그리고 에이전트가 할인에 대한 비즈니스 룰을 문의하였을 때 쇼핑몰 온톨로지서 스캔되어 필요정보가 추출된다. 이 논문에서는 배송비와 비슷한 방식대로 일정한 구매액 이상이 되면 할인을 하는 방식이 적용된다.

5.3 온톨로지와 SWRL로부터의 자료 추출

인터넷에서 상품을 구매하면 오프라인 상점에서 상품을 구매하는 것에 비해 시간을 절약하는 면이 있지만 이외에도 다른 여러 잇점이 있을 수 있다. 그 중 하나는 인터넷 쇼핑몰의 적립금 제도로 인해 얻게 되는 적립금들은 미래의 추가적인 구매에 할인을 받을 수 있는 장점이 있다. 또한 다양한 배송과 할인 혜택은 적절한 상품과 쇼핑몰과의 조합을 찾는다면 추가적인 비용 절감의 효과를 거둘 수 있다. 여기서는 두 번째의 배송과 할인 혜택을 이용해 상품과 쇼핑몰과의 적절한 조합을 찾아 최소의 구매 비용을 찾는 데 초점을 맞춘다.

앞에서 설명한 바와 같이 에이전트는 쇼핑몰 온톨로지와 일반구매 온톨로지를 참조하여 [그림 6]

```

DeliveryRatesRule-1:
  CP:Shoppingmall(?mall) ∧ CP:MallName(?mall, "CDParks") ∧
  CP:purchaseAmount(?mall, ?amount) ∧ swrlb:lessThan(?amount, 20000)
  → CP:deliveryRate(?mall, 1000)
    
```

[그림 5] 배송비 규칙의 예

의 가격표를 작성한다. 필요한 자료로는 쇼핑몰 목록 (부분 ①), 고객이 선택한 제품 목록(부분 ②), 제품의 판매 가격(부분 ③), 구매 수량(부분 ④), 배송비 부과의 준거구매액(DRT), 부과되는 배송비

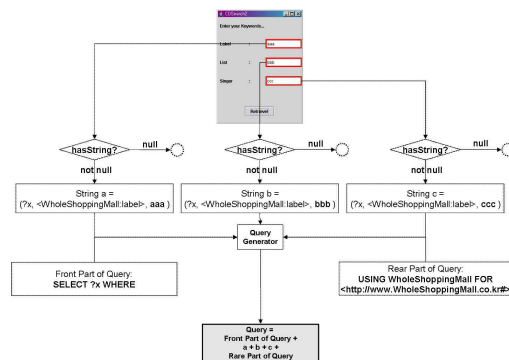
| | CDVMall | HifiTracks | CDPark | LogMusic | OMusic | CDBuy | Quantity |
|--------|---------|------------|----------|----------|----------|----------|----------|
| RETURN | 11000 | 11000 | 10700 | 9500 | 11000 | 10200 | 1 |
| Return | 11000 | 11000 | No Stock | No Stock | No Stock | 10200 | 2 |
| MINA | 11000 | 11000 | 11000 | 9500 | 11000 | No Stock | 3 |
| DRT | 40000 | 25000 | 20000 | 60000 | 25000 | 50000 | |
| DR | 2000 | 1500 | 1000 | 500 | 1500 | 3000 | |
| DCT | 0 | 0 | 25000 | 0 | 0 | 0 | |
| DC | 0 | 0 | 2000 | 0 | 0 | 0 | |

[그림 6] 가격표 스크린샷

(DR), 할인의 준거 구매액(DCT), 적용되는 할인액(DC) 등의 목록(부분 ⑤)이 있고 이에 따른 각 쇼핑몰 별 금액(부분 ⑥)이 필요하다. 이를 위해 각 온톨로지로부터 쿼리를 통해 필요한 정보를 추출하게 되는데 아래에 그 과정을 설명한다.

먼저 고객이 상품을 탐색할 때 상품에 대한 정보는 RDF 형식으로 저장되고 이 정보를 온톨로지로부터 추출할 때 RDQL 쿼리 언어를 사용한다. [그림 7]은 CD를 찾기 위한 쿼리의 한 부분이 어떻게 RDQL로 생성되는지를 보여준다. 고객이 수록 곡 목록, 앨범 명, 가수 명 등 세가지 사항을 입력하면 해당 문자열 “a”, “b”, “c”가 만들어지고 이는 몸체 부분에 결합된다. 여기에 전면부와 후면부가 첨부되어 쿼리가 만들어진다. [그림 8]은 제품의 속성값인 앨범 명, 트랙 목록과 가수 명이 각각 “RETURN”, “temptation”, “Mina”인 제품을 찾는 질의어를 보여주고 있다. 마지막 줄은 쿼리의 네임스페이스가 “WholeShoppingMall”임을 보여주고 있다.

제품을 발견한 후 각 쇼핑몰 온톨로지로부터 제품에 대한 가격 정보를 추출하게 된다. 이때 쇼핑몰 온톨로지에 문의하게 되는 쿼리는 앞에서와 비슷한 쿼리를 사용하게 된다. 단지 다른 점이 있다면 다른 네임스페이스를 사용한다는 것이다. 왜냐하면 각 쇼핑몰은 자체적인 네임스페이스를 가지고 있기 때문이다. 예를 들어 URL이 http://www.CDVMall.co.kr#인 CDV 물은 CDVMall이라는 네임스페이스를 가진다.



[그림 7] 제품탐색을 위한 쿼리를 생성하는 과정

에이전트는 각 고객에 대한 프로필을 보관하고 있고 여기에 고객이 등록한 쇼핑몰을 쉽게 추가하고 삭제할 수 있게 되어있다. [그림 9]는 프로필로부터 쇼핑몰 정보를 추출하는 RDQL 코드이다.

에이전트는 SWRL로더에서 배송비, 배송비를 부과하는데 필요한 준거 구매액, 할인액, 할인에 필요한 준거 구매액 등의 정보를 추출하기 위한 쿼리를 생성한다. 쿼리를 통한 결과가 아무것도 없을 때 에이전트는 SWRL로 표현된 비즈니스 룰을 찾기 위해 일반구매 온톨로지를 탐색한다. 쇼핑몰마다 여러 다양한 비즈니스 룰을 가지고 있으므로 에이전트는 그 중 찾고자 하는 적합한 룰을 찾아야 한다. 배송비 정보를 얻기 위해 우선 “deliveryRate”

라는 술부 단어를 갖고 있는 룰을 찾는다. 그리고 그 룰 중 결과부의 두번째 인자가 0이 아닌 룰을 고른다. 그 룰에서 “purchaseAmount”라는 술부 단어와 “swrlb:lessThan”이란 빌트인 단어를 찾는다. 배송비는 “deliveryRate” 술부의 두 번째 인자이고 배송비 부과의 준거 구매액은 “swrlb:lessThan” 술부의 둘째 인자이다([그림 5] 참조). 할인액과 할인의 준거 구매액도 마찬가지로 추출된다.

6. 한정도메인 제약식을 이용한 최적 의사결정의 도출

고객의 입장에서 가격표에 제시된 정보를 제공 받게 되면 다음 단계로 필요한 것은 추출된 정보를 이용해서 구매 동기에 적합한 의사결정을 하는 것이다. 이 단계에서 에이전트는 수행할 수 있는 쇼핑목표를 제시할 수 있다. 본 논문은 의사결정 과정의 틀을 제시하는 것을 목적으로 하기 때문에 여기서는 모든 가능한 의사결정 목표를 자세하게 구현하지는 않았다. 고객에게 필요한 의사결정 목

표는 일률적으로 제시되기 보다는 각각의 가능한 의사결정 문제를 검토하고 이에 대한 모형과 해법의 존재 여부를 살펴보는 것이 필요하며 여러 프로젝트를 통해 하나의 라이브러리로 완성될 필요가 있다. [그림 10]은 예로써 두 가지 목표를 보여 주고 있다. 하나는 적립포인트를 최대화하는 것이고 또 다른 하나는 전체 구매 비용을 최소화하는 것이다. 후자의 경우 예로 사용될 전체 구매비용 최소화에 필요한 정보들로서 앞의 가격표에 제시되었던 정보들이 구조화되어 요약되어 있다. 즉 고객이 구매비용의 최소화를 구매 목표로 삼는다면 온톨로지 구조에서 보는 것과 같이 구매비용 최소화에 필요한 정보를 필요로 한다. 그리고 이는 [그림 6]에서 제시된 가격표에서 얻어진다. 이 절에서는 [그림 6]에서 얻어진 정보를 이용해 소비자가 최소의 구매 비용으로 원하는 제품을 쇼핑몰에서 구매하기를 원할 때의 문제를 한정도메인 제약식 모형으로 설정하는 것을 보여준다.

[그림 11]은 [그림 6]의 가격표와 [그림 10]의 정보를 이용해서 최소 구매비용 문제를 한정 도메인 제약식 프로그램으로 모형화한 것이다. 고객이 최

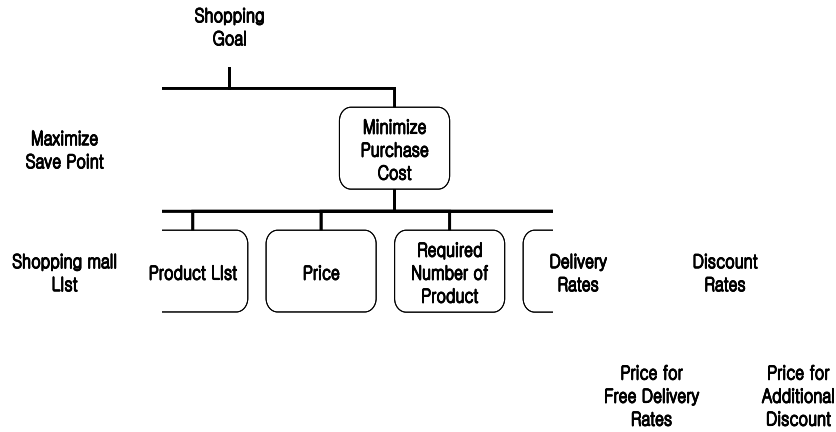
```

1) SELECT          ?x
2) WHERE           (?x, <WholeShoppingMall:label>, "RETURN")
3)                (?x, <WholeShoppingMall:trackListing>, "temptation")
4)                (?x, <WholeShoppingMall:artist>, "Mina")
5) USING          WholeShoppingMall FOR
                  <http://www.WholeShoppingMall.co.kr#>
    
```

[그림 8] 제품탐색을 위한 RDQL 쿼리의 예

| | |
|--------|---|
| SELECT | ?x |
| WHERE | (?z, <http://www.CDSearch.co.kr#password>, "PW") (?z, <http://www.CDSearch.co.kr#identification>, "ID") (?z, <http://www.CDSearch.co.kr#registeredShoppingmall>, ?x)" |
| USING | CDSearch FOR <http://www. CDSearch.co.kr#> |

[그림 9] 쇼핑몰 정보를 추출하는 RDQL 쿼리의 예



[그림 10] 쇼핑목표 온톨로지의 클래스 구조

소의 구매비용이 드는 방식으로 쇼핑물과 제품의 조합을 선택하고자 할 때 가격표에서 보는 바와 같이 세 가지의 비용 요소를 고려한다. 즉 제품 구입 가격, 각 쇼핑물로부터의 배송비 그리고 쇼핑물에서 구입하는 총액에 따른 할인액 등을 고려하여 구매에 따른 총 비용이 최소화되도록 쇼핑물 별로 제품의 종류와 양을 결정하는 것이다. 이를 표현하는 제약식 프로그램 모형을 설명하기 위해 먼저 모형의 변수를 정의하면 다음과 같다.

X_{ij} : 쇼핑물 i 에서 제품 j 의 구매 수량을 의미한다.

TC_i : 쇼핑물 i 에서의 총 구매액을 의미한다.

DC_i : 쇼핑물 i 의 할인액을 나타낸다.

DR_i : 쇼핑물 i 의 배송비를 나타낸다.

TC : 구매자가 지불해야 할 전체 구매비용을 의미한다.

이때 $i=1, \dots, m$ 으로 m 은 고려되는 쇼핑물의 개수를 나타내고 $j=1, \dots, n$ 으로 n 은 구매하고자 하는 제품 종류의 수이다.

먼저 첫째 형태의 제약식으로 변수 X_{ij} 의 값이 한정된 값을 가지므로 이를 한정하는 도메인을 다

음과 같이 정의할 필요가 있다.

$$fd_domain([X_{1j}, X_{2j}, \dots, X_{mj}], 0, N_j)$$

즉 변수 X_{ij} 의 값은 0에서 최대 구매하고자 하는 제품 j 의 총 주문량 N_j 까지의 값 중에서 결정된다. 쇼핑물 i 에서의 구매액을 나타내는 변수 TC_i 는 쇼핑물 i 에서 제품 j 의 가격을 P_{ij} 라 할 때 각 $i=1, \dots, m$ 에 대해서 다음과 같이 정의된다.

$$TC_i \# = P_{i1} * X_{i1} + P_{i2} * X_{i2} + \dots + P_{in} * X_{in}$$

이때 변수 TC_i 의 도메인은 정의되지 않았으므로 초기에 시스템의 최대 도메인으로 설정되고 확산법에 의해 관련 제약식이 처리되면 도메인이 크기가 줄어들게 된다. 세 번째 형태의 제약식으로는 제품 j 의 총 구매량을 계산하는 식으로 각 $j=1, \dots, n$ 에 대해,

$$X_{1j} + X_{2j} + \dots + X_{mj} \# = N_j$$

으로 주어진다. N_j 는 앞에서 정의된 대로 제품 j 의 총 주문량이다. 만약 쇼핑물 i 에서의 구매액이 일정한 금액 DCR_i 을 넘을 때 할인액이 DC_amount_i 만큼 있다면 다음의 제약식에 의해 변수 DC_i 에 제약이 있게 된다.

$(TC_i \# < DCR_i) \# < = > (DC_i \# = 0)$
 $(TC_i \# > = DCR_i) \# < = > (DC_i \# = DC_amount_i)$
 하지만 쇼핑몰 i 에 특별한 할인 정책이 없는 경우 제약식을 통해 변수 DC_i 의 값을 0으로 놓게 된다. 즉,
 $DC_i \# = 0$

의 제약식이 추가된다. 문제에서 제일 다루기 힘든 것은 배송비의 경우이다. [그림 7]에서 본 바와 같이 한 쇼핑몰에서 구매하는 액수가 일정 액수 이상이 되지 않으면 배송비를 추가하게 된다. 즉 쇼핑몰 i 로부터의 구매액이 0이거나 쇼핑몰에서 정한 일정한 액수 DRR_i 이상이 되면 배송비 0, 0과 DRR_i 사이에 있으면 배송비를 DR_amount_i 로 설정하는 제약식은 다음과 같다.

$((TC_i \# = 0) \# \setminus (TC_i \# > = DRR_i)) \# < = >$
 $(DR_i \# = 0)$
 $((TC_i \# > 0) \# \setminus (TC_i \# < DRR_i)) \# < = > (DR_i$
 $\# = DR_amount_i)$

위의 제약식을 만족하는 해를 구하기 위해서는 fd_labeling 제약식을 사용한다. 기본적으로 앞에 설정된 제약식에 의해 제약식 솔버의 내부에서 제약식 네트워크를 구성하게 된다. 그리고 fd_labeling 제약식에 변수의 리스트가 주어지면 각각의 변수에 대해 각 도메인에서 허용되는 값을 변수의 값으로 고정하고 이에 따른 확산법을 적용함으로써 만들어진 제약식 네트워크에서 해 찾기를 시작한다. 따라서 문제에서는 확산법의 시작을 위해 다음과 같은 제약식이 필요하다.

$fd_labeling([X_{11}, \dots, X_{mm}, DC_1, \dots, DC_m, DR_1, \dots, DR_m, TC_1, \dots, TC_m$
 $])$.

해결하려는 문제가 만약 실현 가능해를 찾는 것에 만족한다면 위의 제약식으로 충분하다. 하지만 예에서는 최적해를 찾는 것이 목적이므로, 일단 현재까지 서술된 모든 제약식을 만족하는 해 하나를

발견하는 것에 만족되지 않고 주어진 구매 비용을 최소로 만들기 위해 더 이상의 해가 존재하지 않을 때까지 탐색을 한다. 이를 위해 먼저 각 해에 대해 전체 구매 비용 TC 를 각 쇼핑몰에서의 구매액, 배송비, 그리고 할인액을 이용하여 계산하는 다음의 제약식을 fd_labeling 제약식 이전에 추가한다.

$$\begin{aligned}
 TC \# = & TC_1 + TC_2 + \dots + TC_m \\
 & + DR_1 + DR_2 + \dots + DR_m \\
 & - DC_1 - DC_2 - \dots - DC_m
 \end{aligned}$$

구매 비용의 최소화는 fd_minimize 제약식을 이용한다. 즉 첫째 인자로 실현 가능해를 구하는 제약식을 주고 둘째 인자로 구매 비용 값을 보여주는 변수를 설정한다. 따라서 원하는 제약식을 다음과 같이 설정한다.

$L = [X_{11}, \dots, X_{mm}, DC_1, \dots, DC_m, DR_1, \dots, DR_m, TC_1, \dots, TC_m],$
 $fd_minimize(fd_labeling(L), TC)$.

[그림 6]에서 제시된 예제를 풀기 위한 프로그램은 [그림 11]에서 보여주고 있다. 제약식 네트워크를 구성하는 제약식을 한 곳에 묶어 shopProb으로 표현하고 있고 에이전트가 shopGoal을 호출하면 shopProb가 먼저 호출되어 제약식 네트워크가 구성된다. 그리고 fd_minimize 제약식을 호출하여 확산법을 시동하고 이를 통해 최소의 구매 비용과 이를 실현하는 해의 값이 변수 TC 와 변수 리스트 L 을 통해 도출된다. 이와 같은 모형을 gprolog의 한정도메인 제약식 프로그램 솔버에 입력하여 풀었을 때, 쇼핑 에이전트는 [그림 6]의 예에서 전체 구매비용을 최소화하기 위하여 “HotTracks”에서 “Rendezvous” 2단위와 “MINA3” 1단위를 구매하고 “CDPark”에서 “RETURN” 1단위와 “MINA3” 2단위를 63,700의 비용에 구매하는 것이 최적임을 사용자에게 알려주게 된다.

[그림 11] 최소 구매비용 문제의 제약식 프로그래밍 모형

```

shopProb(L, TC) :-
    fd_domain([X11, X21, X31, X41, X51, X61], 0, 1),
    fd_domain([X12, X22, X32, X42, X52, X62], 0, 2),
    fd_domain([X13, X23, X33, X43, X53, X63], 0, 3),
    fd_domain([DC1, DC2, DC3, DC4, DC5, DC6], 0, 20),
    fd_domain([DR1, DR2, DR3, DR5, DR6], 0, 30),
    TC1 #= 110 * X11 + 110 * X12 + 110 * X13,
    TC2 #= 110 * X21 + 110 * X22 + 110 * X23,
    TC3 #= 107 * X31 +          110 * X33,
    TC4 #= 98 * X41 +          98 * X43,
    TC5 #= 110 * X51 +          110 * X53,
    TC6 #= 102 * X61 + 102 * X62,
    X32 #= 0, X42 #= 0, X52 #= 0, X63 #= 0,
    X11 + X21 + X31 + X41 + X51 + X61 #= 1,
    X12 + X22 + X32 + X42 + X52 + X62 #= 2,
    X13 + X23 + X33 + X43 + X53 + X63 #= 3,
    DC1 #= 0, DC2 #= 0,
    (TC3 #< 250) #<=> (DC3 #= 0),
    (TC3 #>= 250) #<=> (DC3 #= 20),
    DC4 #= 0, DC5 #= 0, DC6 #= 0,
    ((TC1 #= 0) #\| (TC1 #>= 400)) #<=> (DR1 #= 0),
    ((TC2 #= 0) #\| (TC2 #>= 250)) #<=> (DR2 #= 0),
    ((TC3 #= 0) #\| (TC3 #>= 200)) #<=> (DR3 #= 0),
    ((TC4 #= 0) #\| (TC4 #>= 600)) #<=> (DR4 #= 0),
    ((TC5 #= 0) #\| (TC5 #>= 250)) #<=> (DR5 #= 0),
    ((TC6 #= 0) #\| (TC6 #>= 500)) #<=> (DR6 #= 0),
    (TC1 #> 0 #/\ TC1 #< 400) #<=> (DR1 #= 20),
    (TC2 #> 0 #/\ TC2 #< 250) #<=> (DR2 #= 15),
    (TC3 #> 0 #/\ TC3 #< 200) #<=> (DR3 #= 10),
    (TC4 #> 0 #/\ TC4 #< 600) #<=> (DR4 #= 25),
    (TC5 #> 0 #/\ TC5 #< 250) #<=> (DR5 #= 20),
    (TC6 #> 0 #/\ TC6 #< 500) #<=> (DR6 #= 30),
    TC #= TC1 + TC2 + TC3 + TC4 + TC5 + TC6
        + DR1 + DR2 + DR3 + DR4 + DR5 + DR6
        - DC1 - DC2 - DC3 - DC4 - DC5 - DC6,
    L = [X11, X21, X31, X41, X51, X61,
        X12, X22, X32, X42, X52, X62,
        X13, X23, X33, X43, X53, X63,
        DC1, DC2, DC3, DC4, DC5, DC6,
        DR1, DR2, DR3, DR4, DR5, DR6,
        TC1, TC2, TC3, TC4, TC5, TC6].

shopGoal :-
    shopProb(L, TC),
    fd_minimize(fd_labeling(L), TC),
    write(TC).
    
```

7. 결론 및 한계와 앞으로의 연구 방향

이 논문은 인터넷에서 최적의 구매 의사결정을 돕기 위해 CD 쇼핑물의 경우를 중심으로 의미 웹 기반의 제약식 프로그래밍을 이용한 지능형 에이전트를 제시하였다. 인터넷에서 의미 웹 구조를 전제로 에이전트를 위한 온톨로지 구현에 대해 살펴 보았고 각 온톨로로부터 데이터가 추출되고 이것이 최적의 구매 의사결정을 위해 한정도메인 제약식 프로그래밍을 이용하여 의사결정 모형의 설정에 어떻게 사용되는지 보았다. 그리고 설정된 모형을 한정도메인 제약식 프로그래밍 솔버를 이용하여 해를 구함으로써 에이전트가 구매 의사 결정자에게 최적의 해를 제시하는 의사결정의 틀을 보여주었다.

본 연구로는 완벽하게 실제 상황에 활용 가능한 에이전트를 구현하는데 아직은 미흡한 점이 많다. 즉 현재의 구현은 비즈니스의 틀로써 두 가지 사항만을 고려하였으나 실제 상황에서는 훨씬 많은 고려 사항이 필요하다. 또한 구매 시 의사결정의 목표로써 총 구매액을 최소화하는 것으로 에이전트의 구조를 설명하였지만 추후의 연구에서는 구매 시에 어떠한 목표들이 있고 또 어떤 목표들이 실제로 중요하게 작용하는지를 실증적으로 연구할 필요가 있다. 예를 들어 소비자는 일반 쇼핑물에서 운용하는 적립금 제도를 통한 장기적인 할인 액수를 최대화하기를 원할 수도 있다. 또한 제시된 구매 목표를 달성하는 의사결정을 위해 한정도메인 제약식 프로그래밍 모형을 사용하였는데 이 부분은 프로토타입만이 구현이 된 상태이다. 특히 여러 상황과 조건의 변화에 따라 유연하게 모형을 생성해줄도록 하는 모형 언어에 대한 부분은 이 논문에서는 구현되지 않았다. 하지만 최근에 제시된 SWRL하에서 제약식과 인터페이싱을 위한 확

장 언어가 그 단계를 보완해 줄 수 있다. 마지막으로 서론에서 제기한 것처럼 제안된 에이전트가 많은 실제 문제를 완전하게 해결하기 위해서는 실제 문제를 해결할 수 있는 여러 형태의 제약식 솔버의 구현이 필요하다.

참고 문헌

- [1] Antoniou, G. and F. van Harmelen, "Web Ontology Language: OWL", *Handbook on Ontologies in Information Systems*, Springer, (2003), 76-92.
- [2] Bassiliades, N. and P.M.D. Gray, "Colan: A Functional Constraint Language and Its Implementation", *Data & Knowledge Engineering*. Vol.14(1994), 203-249.
- [3] Berners-Lee, T.(ed.), *Weaving the Web*, Orion, 1999.
- [4] Codognet, P. and D. Diaz, "Compiling Constraints in clp(FD)", *Journal of Logic Programming*, Vol.27, No.3(1996).
- [5] Decker, S., M. Erdmann, D. Fensel and R. Studer, "Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information", In R. Meersman et al. (eds.), *Semantic Issues in Multimedia Systems*, Kluwer Academic Publisher, Boston, 1999.
- [6] Diaz, Daniel and P. Codognet, "Design and Implementation of the GNU Prolog System", *Journal of Functional and Logic Programming*, Vol.2001, No.6(2001).
- [7] Fruehwirth, Thom and S. Abdennadher, *Essentials of Constraint Programming*, Springer-Verlag, Berlin, 2003.
- [8] Gray, Peter, Kit Hui and Alun Preece,

- “An Expressive Constraint Language for Semantic Web Applications”, IJCAI01 Workshop on e-business and the intelligent web, August 5, 2001.
- [9] Greenwald, A. R. and J. O. Kephart, “Shopbots and Pricebots”, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, (1999, July 31 ~ August 06), 506-511.
- [10] Grigoris Antoniou and Frank van Harmelen, “Semantic Web Primer Chapter 1. The Semantic Web Vision”, available from <<http://www.ics.forth.gr/isl/swprimer/presentations/Chapter1.ppt>>.
- [11] Gruber, T.R., “A translation approach to portable ontologies”, *Knowledge Acquisition*, Vol.5, No.2(1993), 199-220.
- [12] Hahn, U. and S. Schulz, “Building a very large ontology from medical thesauri”, *Handbook on Ontologies in Information Systems*, Springer, 2003, 133-150.
- [13] Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosf and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, Acknowledged W3C Member Submission, NRC 48058, May 21, 2004.
- [14] Maedche, Alexander and Steffen Staab, “Ontology Learning”, *Handbook on Ontologies in Information Systems*, Springer, (2003), 173-190.
- [15] Menczer, F., W. N. Street, N. Vishwakarma, A. E. Monge and M. Jakobsson, “IntelliShopper: a proactive, personal, private shopping assistant”, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, part 3, ACM Press, 2002.
- [16] Miller, E, “An Introduction to the Resource Description Framework”, *Bulletin of the American Society for Information Science*, Vol.25, No.1(1998).
- [17] Mitra, Prasenjit and Gio Wiederhold, “An Ontology-Composition Algebra”, *Handbook on Ontologies in Information Systems*, Springer, (2003), 93-113.
- [18] Noy, N. F., M. Sintek, S. Decker, M. Crubézy, R. W. Ferguson and M. A. Musen, “Creating Semantic Web Contents with Protégé-2000”, *IEEE Intelligent Systems*, Vol.16, No.2(2001).
- [19] Pack, Thomas, “Intelligent shopping agents”, *Link-up*. Medford, Vol.18, No. 2(2001, Mar~Apr), 16.
- [20] Preece, Alun, Stuart Chalmers, Craig McKenzie, Jeff Z. Pan and Peter Gray, “Handling Soft Constraint in the Semantic Web Architecture”, WWW2006, (2006, May), 22-26.
- [21] Prud’hommeaux, E. and A. Seaborne, “SPARQL query language for RDF”, Technical report, World Wide Web Consortium (2005).
- [22] Sproule, S. and N., Archer, “A Knowledgeable Agents for Search and Choice Support in E-commerce: A Decision Support Systems Approach”, *Journal of Electronic Commerce Research*, Vol.1, No.4(2000), 152-165.
- [23] Silverman, B., M. Bachann and K. Al-Akharas, “A Markov Decision Processing Solution to Natural Language Querying of Online e-Commerce Catalogs: The EQUI search Agent”, available at <<http://www.seas.upenn.edu/~barryg/mdp.pdf>>.

- [24] Sure, York, Steffen Staab and Rudi Studer, "On-To-Knowledge Methodology", Handbook on Ontologies in Information Systems, Springer, (2003), 117-132.
- [25] Wen-Shan, Lin, "Knowledge modelling and its applications for an intelligent shopping agent", Proceedings of 2005 International Conference of Machine Learning and Cybernetics, (2005), 5584-5589.
- [26] Wooldridge M., "An Introduction to Multi-Agent Systems", John Wiley & Sons Publishers, available from <<http://www.csc.liv.ac.uk/~mjw/pubs/imas/>>.

Abstract

Intelligent Shopping Agents Using Finite Domain Constraint under Semantic Web

Hak-Jin Kim^{*} · Myung Jin Lee^{**}

When a consumer intends to purchase products through Internet stores, many difficulties are met because of limitations of the current search engines and the current web structure, and lack of tools supporting decision-makings. This paper raises an Internet shopping problem and proposes a framework of decision making process to settle it with an intelligent agent based on Semantic Web and Finite Domain Constraint. The agent uses finite domain constraint programming as modeling and solution methods for the decision problem under the Semantic Web environment.

Key words : Agent, Semantic Web, Constraint Programming, Finite Domain Constraint Programming

* Dept. of Business Administration, Yonsei University

** Dept. of Industrial & Information Engineering, Yonsei University