

웹 에이전트 기술의 전자상거래 응용

김대호¹⁾, 박상호²⁾

1. 서 론
 2. 웹 에이전트(Web Agent) 기술
 3. BuyingAgent와 SellingAgent의 Model단계
 4. 에이전트 개발도구 : AgentBuilder의 활용
 5. 결론 및 향후 연구과제
- <참고문헌>

1. 서 론

오늘날 정보통신기술의 발달로 인터넷의 이용이 급증함에 따라 전자상거래의 이용 또한 증가하고 있다. 이들 사용자는 인터넷상에서 수많은 사이트를 돌아다니며, 자신이 원하는 정보를 찾으려 하는데, 시간에 여유를 갖는 사람들만이 많은 곳을 돌아다니며 찾을 것이다. 물론 시간의 여유를 갖는 사용자도 되도록 보다 손쉬운 방법으로 정보를 찾으려 할 것이다. 특히 치열한 경쟁사회에서 정보에 대한 마인드는 20세기 경영의 키워드로 부각되었고 기업과 개인생활에서 업무의 효율성과 신속정확성을 요구한다.

많은 정보를 취급하는 인터넷 사용자 특히 전자상거래 상에서 사용자들은 자신이 원하는 정보를 보다 쉽고 빠르게 찾기를 원한다. 만약 이러한 정보의 검색하고 사용자들의 업무를 대행해주는 소프트웨어가 있으면 보다 편리할 것이다. 이러한 것을 대행해주는 것이 에이전트이며, 지금 현재 이러한 소프트웨어는 많이 개발되어 운영되고 있으며 정보공유를 통해서 수많은 연구소나 회사에서 배포하고 있다. 그리고 지속적인 업그레이드로 그 기능이 발전되고 있고 새로운 에이전트를 개발 중에 있다. 이러한 것들은 현대기업의 경영측면에서도 문제해결을 보다 과학적으로 해결하여 경쟁전략에서 경쟁우위를 이룰 수 있을 것이다.

전자상거래와 에이전트에 관한 일반적인 특징을 통해서 사용자가 정보를 찾는데 판매자들이 제품정보에 대해 디스플레이한 장소를 찾아주는 에이전트에 관해서 논하고자한다. 즉 전자상거래 상에서 운영할 수 있는 기술을 구현하는데 필요한 방법에 대하여 살펴보고, 그리고 전자상거래 상에서 에이전트를 구현하여 운영할 수 있는 에이전트를 구현할 수 있는 툴들이 있는데 그중에서 AgentBuilder를 이용해 에이전트 구현을 소개하고자 한다

2. 웹 에이전트(Web Agent)기술

1) 목원대학교 정보관리학부 경영정보전공 조교수
2) 목원대학교 대학원 석사과정

2-1. 웹 에이전트

에이전트(Agent)라는 단어의 원래의 뜻은 ‘대리인’을 뜻하는데, 이는 인터넷이나 다른 프로그램에서 다양하게 정의되는데, 사용자를 대표 또는 대신하여 사용자가 해야 할 작업을 자동으로 수행하는 ‘소프트웨어’라고 할 수 있다. 이는 하드웨어(H/W) 에이전트인 로봇과는 대별되는 인간의 대리개념으로 사용자들의 행동양식을 관찰하고 학습하여 정보공간에서 사용자를 대표하고 학습된 사용자의 행동양식을 기반으로 사용자가 해야 할 작업을 자동으로 수행해 주는 소프트웨어라 불릴 수 있다. 이렇게 사람이 환경에 적응하는 것처럼, 에이전트 역시 사람에 적응해야 한다. 사람은 계속되는 시행착오를 거치면서 자신이 처한 환경을 흉내내면서 학습하게 된다. 이와 동일하게 에이전트 역시 사람의 행동을 관찰하고 학습하는 기능, 즉 자신이 처한 환경을 학습하는 기능을 갖는데 이를 ‘에이전트 엔진(Agent Engine)’라 한다.

이런 에이전트는 초기에는 관심분야가 비슷한 사람들끼리 접할 수 있는 에이전트로 정보를 여과하는 기능을 하며, 반복적 행위를 자동화하고자 하였다. 최근에는 DAI의 장점을 계승하여 경영업무를 예측하고 미리 행동할 수 있는 에이전트가 개발되어 전자상거래에 활용되고 있다.

2-2. 웹 에이전트 도입의 필요성

현대 기업경영에 있어 기업들은 경영전략을 세우는데 있어 정보기술을 기반으로 전략을 세우는데, 기업정보시스템구축의 한 부분으로써 에이전트기반 시스템을 구축하는데 필요한 에이전트기술에 대해 알아보하고자 한다.

에이전트기반시스템을 구축하는 두 가지 접근방법이 있는데, 단독에이전트를 사용할 수 있고, 이들의 결합인 다중에이전트시스템을 구현할 수 있다. 단독에이전트는 단지 사용자와 함께 의사소통을 하고 에이전트기반프로그램을 구현하기 위해 요구되는 모든 기능을 제공한다. 즉 자체적으로 어떤 일을 수행할 수 있는 능력과 바이어스 등을 지닌 에이전트이다. 다중에이전트시스템은 특별한 시스템프로그램에 의지해서 단독에이전트가 이루는데 어렵고 불가능한 어떤 일들을 이루기 위해 여러 에이전트로 협력하는 전사적 시스템이다.

이러한 에이전트시스템을 구축하는 것이 왜 중요한가? 기업들은 기업간, 기업과 고객간, 기업 내에서 기업을 운영해야 하기에 전사적 응용프로그램을 개발해야 한다. 이를 위해서는 기업들은 더더욱 그들의 공급자와 고객에게 그들의 정보시스템을 통합하고자 한다. 그리고 새로운 시스템은 WWW와 분산된 객체기술을 사용하는 통합된 정보관리시스템으로 복합적 조직과 복합적인 응용프로그램 플랫폼을 연결할 수 있어야 한다. 이렇게 함으로써 기업의 궁극적인 목표를 성취하는데 있어 보다 효율적이다.

그러면 왜 에이전트들을 구축하는데 어려운가? 에이전트와 에이전트기술은 인공지능과 컴퓨터공학 분야에서 활발한 연구분야를 가지고 있다. 다수의 대학에서 지능형 소프트웨어에이전트를 개발했고, 많은 기업들은 광범위하게 전문화된 일들을 수행하는 소프트웨어에이전트 능력을 개발했다. 어쨌든 이 에이전트들 각각은 특별한 어플리케이션에 대해 수작업으로 했기 때문에 지능형 소프트웨어에이전트를 구축하는데 지식표현, 추론, 네트워크 커뮤니케이션 방법과 프로토크 등과 같은 진보된 기술을 이해를 하는데 어려움과 많은 시간을 소비한다. 그리고 복잡한 어플리케이션은 종종 기계학습과 기계계획기술(mechanical planning technique)에서 전문가적 지식기술을 요구한다.

이러한 에이전트를 개발하기 위해 기본적으로 소프트웨어에이전트 개발자들은 에이전트기반

어플리케이션들을 개발할 때, 그들에게 도움을 줄 수 있는 툴의 집합을 요구하는데, 그 툴은 소프트웨어개발자가 어플리케이션영역, 즉 정형적으로 그 영역에 적절한 개념, 관계, 객체들을 인지하고 기술하고, 영역 안에서 에이전트(들)의 행태를 명확히 하는 영역을 분석하는데 도움을 줄 수 있는 것들이 요구된다. 또한 에이전트의 집합을 명확히 하고 에이전트들간에 메시지와 메시지프로토콜을 분석하여 명확히 하고 에이전트의 행동을 수행하고 평가하는 툴을 요구한다. 이런 필요성을 인식하여 에이전트 시스템을 개발하는데 도움이 되는 툴을 살펴보겠다.

현재 많은 툴들이 있는데 그러한 툴들 중의 하나가 Reticular System, inc.의 AgentBuilder이며, AgentBuilder는 두 개의 중요한 요소로 구성되어 있다. 그것은 개발 툴, 즉 에이전트문제영역을 분석하는 것과 에이전트 행태를 명확히 하는 에이전트프로그램을 생성하는데 사용된 것과 런-타임(run-time) 수행환경, 즉 이들 에이전트프로그램을 수행하는데 높은 성과를 갖는 에이전트엔진을 제공한다는 것이다. AgentBuilder를 사용하여 구축되는 에이전트는 KQML(Knowledge Query & Manipulation Language)을 사용하여 커뮤니케이션을 하고 KQML로 정의된 수행 문(statement)을 지원한다. 그리고 AgentBuilder 툴 킷과 런-타임엔진은 자바프로그래밍언어를 사용하여 구현되었다.

2-3. 웹 에이전트의 특징

소프트웨어에이전트는 다른 소프트웨어 또는 인적에이전트(HumanAgent)와 함께 커뮤니케이션 능력과 환경을 인지하고 감시할 수 있는 능력을 가지는데, 정적 소프트웨어 에이전트는 서로 독립적으로 운영되고 때로는 동적 소프트웨어 에이전트로 상호 협력하여 업무를 처리하기도 한다.

일반적인 소프트웨어와 구별되는 특징을 지닌 웹에이전트의 특징과 속성을 살펴보면, 먼저 지능을 들 수 있다. 지능(Intelligent)이란 어떤 일을 대할 때 전문적인 지식을 가지고 대할 때 준다면 그 소프트웨어는 인공지능분야의 전문가시스템(ES)과 다를 바 없다. 그 예로 전문가시스템도 지능형 에이전트의 예이다.

자율성(Autonomy)을 들 수 있는데, 사람이나 다른 에이전트의 간섭 없이 행동할 수 있으며, 자신의 행동과 내부상태에 대한 제어권을 가진다. 그리고 에이전트는 웹 상에서 이동(Mobile)이 가능하다는 것이다. 이런 자율성 등의 특성을 지니고서 다른 호스트 컴퓨터로의 이동이 필요하다. 즉 초기에 부여받은 임무를 수행하기 위해 자신이 생성된 호스트를 벗어나 네트워크상의 다른 호스트에서 작업을 수행하거나 임무수행에 필요한 정보를 구하기 위해 네트워크 사이를 이동할 수 있는 능력을 말한다.

이와 같은 에이전트는 그 임무를 수행하기 위해 다른 에이전트와 커뮤니케이션하기 위해서는 친화성이 필요하다. 이때 필요한 기능이 사교능력으로 사람과의 상호작용을 주로 다루는 분야를 인간과 컴퓨터의 상호작용(Human Computer Interface)로 알려져 있다. 에이전트가 제기능을 발휘하기 위해 대화와 협력(Collaborative)이 필요한데, 에이전트간의 통신을 위한 시도는 주로 ACL(Agent Communication Language)로 나타나고 있다.

마지막으로 에이전트는 적응성(adaptivity)을 지녀야되는데, 자신이 처한 환경에 능동적으로 대처할 수 있다. 즉 자신이 처한 환경에서의 경험을 통해 환경을 학습하고 이러한 지식을 바탕으로 환경변화에 대처한다는 의미이다.

여기서는 웹 상에서 에이전트를 운영하는데 있어, 지능형 에이전트가 바로 웹에이전트에

서 필요한 특징들인데, 위에서 상술한 특징은 ES도 몇몇 포함하고 있지만, 전문가시스템이나 일반 에이전트와는 다른 에이전트이다. 그리고 DAI(Distributed Artificial Intelligent)의 장점을 계승하여 적극 활용하고 있다.

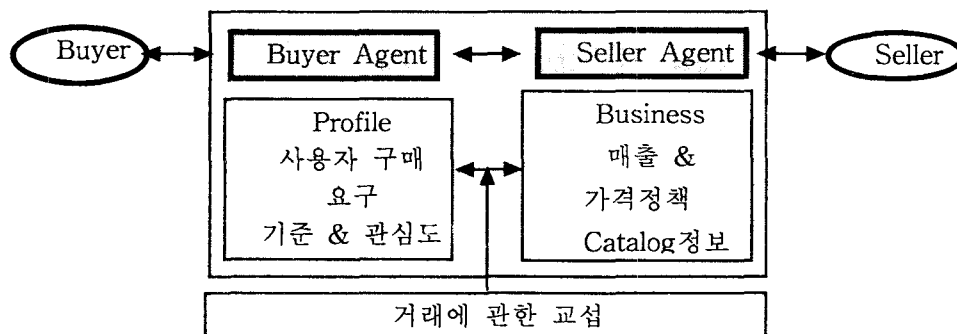
2-4. 웹 에이전트의 응용분야

기업활동이나 개인의 일상업무, 즉 이메일을 주고받고, 걸려온 전화를 받고, 연구자료를 수집 분석하고 스케줄 상황을 수시로 체크하는 등에서 에이전트를 기본적으로 응용할 수 있다. 이런 에이전트의 응용분야는 크게 이벤트모니터, 업무흐름지원, 인터넷자료수집과 검색에이전트가 있다. 현재 웹 상에서 응용되고 있는 것을 살펴보면, 웹 서치 에이전트(Web Search Agent)로서 사용자의 탐색엔진을 대신하는 로봇을 의미하고, 정보여과(Information Filtering) 에이전트는 사용자의 개인적인 관심에 따라 인터넷의 정보를 필터링 하여 주는 시스템을 의미한다. Off-Deliver 에이전트는 개별화된 온라인 정보를 사용자가 원하는 방법에 따라 사용자의 PC로 전송하여 주는 시스템을 의미하고, 공지(Notification) 에이전트는 여러 정보들 중 특정 사용자 개인에게 중요한 이벤트를 통보하여 주는 시스템을 의미한다. 서비스 에이전트는 특정 서비스 사용자의 요구사항과 그 서비스의 내용이 갖는 특징을 상호 비교하여 주는 정보 중개로서의 역할을 수행하는 시스템을 의미하고, 웹사이트 에이전트는 MUD Site등에서 활용되는 것으로서, 하나의 웹사이트에서 이곳에 방문한 사용자들에게 전자적인 모습의 호스트로서의 역할을 수행할 수 있는 시스템을 의미한다. 에이전트는 이런 다양한 역할을 수행하며 운용되고 있다. 이러한 것들은 개인활동이나 업무를 도와주는 컴퓨팅 환경의 일례들이다.

전자상거래 상에서의 에이전트 구축/운영사례를 보면, 전자상거래에서는 모든 것을 자신이 직접 처리해야한다. 그러나 전자상거래 환경에서 정제되지 않는 정보를 소비자에게 제시할 경우 실세계와 마찬가지로 소비자는 물건을 구매하기 위해 컴퓨터 앞에서 엄청난 노력을 들여야 할 것이다. 이런 현상을 '정보 과부하'라 하는데, 사용자가 해야할 작업을 자동으로 수행해 주는 소프트웨어에이전트를 통해 해결할 수 있다. 외국의 에이전트 운영 사이트 사례들이 많이 운영되고 있다.

3. BuyingAgent와 SellingAgent의 Model단계

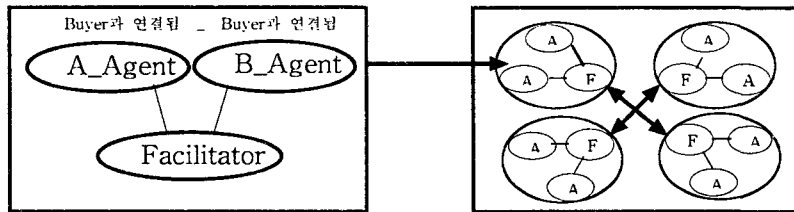
3-1. 에이전트의 주요기술



<그림-1> 에이전트 핵심기술 기본모델

에이전트는 기본적으로 정보의 발견, 커뮤니케이션, 온톨로지, 러거시(Legacy)시스템, 추론과 조정 그리고 모니터가 있다. <그림-1>은 기본적인 에이전트의 핵심기술로써 하나의 에이전트 안에 Buyer 에이전트와 Seller 에이전트로 구성되어 사용자의 구매요구 기준 및 관심도와 판매자의 매출 및 가격정책 등 카탈로그의 양쪽 정보를 교섭하며, 하나의 에이전트의 기능을 수행한다.

사업의 목표와 고객의 이해관계를 연결하여 교섭하는 에이전트이다. 사업의 목표 면에서는 고객의 충실도, 매출성장, 고객의 욕구 이해, 고객에 대한 피드백 등을 들 수 있다. 고객의 이해 면에서는 서비스의 차별화, 개별화, 정보에 대한 쉬운 접근, 유사한 라이프스타일을 가진 사람들의 모임, 고객들의 추천 그리고 비공식적인 모임의 지속적인 면을 연결해 준다.



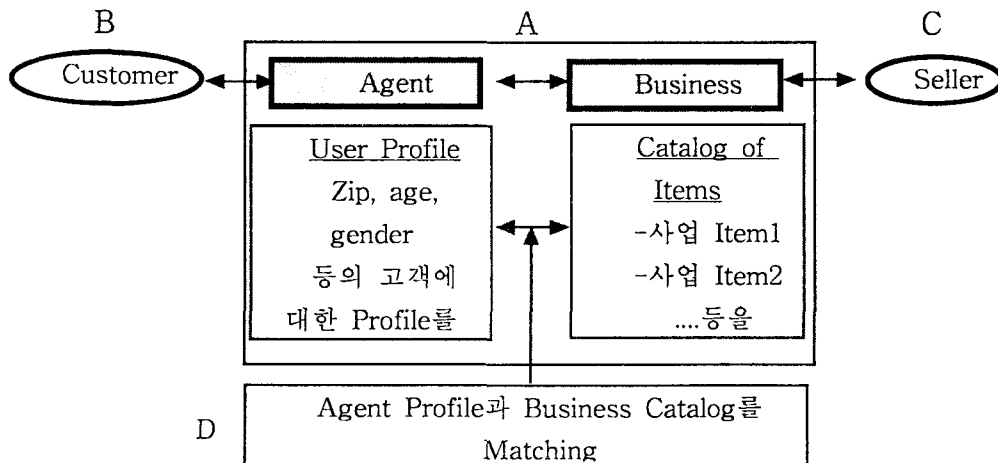
<그림-2>직접 에이전트와 조정 에이전트

직접 에이전트는 에이전트 자체 내에서 촉진자(Facilitator)를 통해 독립적으로 운영할 수 있는 에이전트이고 KIF(Knowledge Interchange Format)를 통해 메시지를 교환하면 된다. 그러나 이종에이전트간에서는 촉진자가 에이전트 촉진자간에 서로 다른 에이전트이기에 온톨로지된 KQML을 사용하여 메시지를 교환한다.

3-2. 6단계 구매행태모델(Buying Behavior Model)

(1) 필요식별단계(Need Identification)

이 단계에서는 구매자들에게 어떤 것(정보, 제품)이 적합한 요구인지 또는 적합치 않은 요구인지 인식하는 것을 특성화한다. 즉 제품정보를 통해 구매자는 자극을 받는다. 인터넷을 검색하는 동안 구매하고자 하는 상품에 대해 인식하는 단계이다.



<그림-3> 에이전트 수행 모델

이를 위해 이용자의 프로파일(Profile)이 매우 중요시되는데 이 프로파일을 적극적으로 관리할 수 있는 관리자가 필요하다. 이것을 에이전트가 수행할 수 있다.

여기서 에이전트의 역할은 이용자의 개인정보를 바탕으로 사용자가 관심을 가질만한 상품정보를 적극 추천하기도 하고 비슷한 취향을 갖는 다른 이용자의 최근 구매정보를 통하여 최신물품을 제공하기도 한다.

이런 종류로 가장 오래되고 단순한 소프트웨어에이전트로는 'monitors'가 있다. 사용 예로는 주식시장과 전자상거래 사이트들이 있다. 그 예로 Amazon.com 과 Firefly.com이 개발중이다.

(2) 제품브로커링 (Product Brokering)

이 단계에서는 무엇을 살 것인가 결정하는데 돕는 정보검색을 포함하고 구매자의 구매기준이 되는 기본적인 제품선택의 평가를 포함한다. <그림-3>의 D에서 검색된 상품정보의 기준평가를 통해 무엇을 구매할 것인가를 결정하는 것이다. 즉 사용자의 의사결정에 필요한 각종정보를 제공한다.

이러한 사용자의 관심/요구에 맞게 구축한 에이전트의 대표적 사례로는 Personallogic(www.personallogic.com), Firefly(www.firefly.com), 그리고 Tete-a-Tete(ecommerce.media.mit.edu)등을 들 수 있다. Personallogic은 사용자가 만족할 수 있게 여러 가지 선택을 함으로써 결과로 사용자에게 맞는 상품에 대한 정보가 나온다. 즉 주어진 영역 내에서 원하지 않는 상품에 대해서는 필터링 한다. Firefly는 개인의 정보를 수집할 수 있게 해놓고 그들간에 공동체를 구성하게 에이전트를 만들었다. 그리고 개인의 사생활(Privacy)을 보호하도록 해 놓았다. Tete-a-Tete는 다중속성효용이론(multi-attribute utility theory)을 기반으로 복잡한 상품을 추천하여 비교할 수 있는 기술을 사용한다. 또한 Merchant Brokering과 Negotiation단계에서 Buyers와 Sellers를 보조한다.

여기서 에이전트의 역할은 사용자의 직접적인 요구나 요구발생단계의 에이전트로부터의 자발적인 검색요구에 대응하여 인터넷상에서 정보를 검색한다. 이 과정에서 기존의 정보검색용 엔진을 그대로 사용하는 것으로는 한계가 있다. 즉 검색엔진의 지속적인 학습이 필요하다.

상품에 대한 의사결정 지원에 대한 기술로써 규칙기반권고(Rule-based recommendations), 그리고 고객구매행위 안에서 패턴을 찾기 위해 데이터마이닝(Data-mining) 기술사용, 그리고 제약기반여과(Constraint-based filtering) 등이 소개되고 있다.

(3) 머천트 브로커링 (Merchant Brokering)

누구로부터 구매할 것인가를 결정하는데, 즉 도움이 되는 상인명세(Merchant-Specific)와 전 단계(Product Brokering)의 것을 결합한다. 그리고 그 역할은 다양한 형태의 인터페이스를 가지고 있는 쇼핑물의 분석을 통해 사용자의 요구를 각 쇼핑물이 규정하고 있는 입력형식으로 자동 변환하고 검색결과를 통합하고 재구성하여 비교구매 등이 가능하도록 한다.

이러한 에이전트로는 <그림2>의 D에서 에이전트가 다른 하위에이전트(Subagent)들 안에 있는 정보를 찾아 이용자에게 정보를 제공한다. 운영 예로 Bargainfinder, Jango, AuctionBot, tete-a-tete, marketmaker, 그리고 auctionweb 등이 있다.

(4) 교섭협상 (Negotiation)

이 단계는 거래의 조건을 어떻게 해결할 것인가에 관한 것이다. 가격 또는 거래의 다른 조건에 맞는 교섭을 하는 단계이다. 여기서 이용자는 교섭을 통해 만족의 극대화를 추구한다. 이는 기업간 거래형태인 B-to-B거래가 많이 포함된다.

여기서 에이전트의 역할로써 교섭은 일상적인 상행위이다. 즉 구매자는 싸게 판매자는 좀더 비싸게 팔려고 그에 맞는 여러 가지를 정보라는 형태로 입수하면서 교섭대상을 결정하고 각 대상별로 다양한 새로운 교섭전략을 세우는 것이 요구된다. 이 단계도 경매에이전트로 많이 개발된 것을 볼 수 있다. 운영 예로는 AuctionBot, tete-a-tete, makermaker, auctionweb 등이 있다.

(5) 구매 및 배달 (Purchase & Delivery)

이 단계는 교섭의 끝에서 발생하는데, 사용자가 만족한 상품에 대해 구매를 하는 단계이다. 여기서는 상품에 대한 지불능력과 그것을 인도하는데 따르는 여러 가지 선택이 따르게 된다. 즉 이 단계는 구매한 상품에 대해 지불에 대한 문제로써 현재 중요한 이슈가 되고 있고, 아직 이것을 대행할 에이전트는 구현되지 않고 있지만, 여러 가지 법적 문제가 해결되어 인증 등과 같은 문제들이 해결되면 곧바로 운영 될 수 있을 것이다.

여기서 에이전트의 역할은 인터넷을 통해 거래상대에게 이동하거나 상대의 에이전트를 불러와 실질적인 계약을 수행한다. 이때 에이전트는 계약상황이나 결과를 전자우편이나 전화, 팩스 등을 통해 알려준다.

(6) 서비스와 평가 (Service & Evaluation)

이 단계는 구매후의 단계로서 제품과 고객에 대한 서비스 그리고 전체구매 경험과 의사결정에 따르는 만족의 상태를 평가하는 단계이다. 또한 구매자에 대한 정보를 수집하여 다음에 상품 구입 시 구매자의 요구를 만족시킬 수 있다.

여기서 에이전트의 역할은 계약완료 후 실제로 물품 배달여부의 확인과 물품의 이상여부를 확인한다. 또한 정기적으로 사용자로부터 물품에 대한 평가를 받아 물품과 거래 상대자에 대한 데이터베이스를 구축하여 차후의 거래 시 품목이나 브랜드, 취급업자의 선택이나 평가에 이용한다. 즉 서비스부분으로써 고객의 요구에 지속적으로 만족시킬 수 있는 에이전트의 구현 부분이다.

지금까지는 구매행태모델의 6단계를 살펴보았는데, 이와 비슷하지만 5 단계 모델로 보는 경우가 있다. 이는 6단계 모델 중 두 단계를 하나의 단계로 보았다. 다음으로 에이전트 개발도구로써 AgentBuilder를 활용하여 에이전트 개발에 대해 살펴보겠다.

4. 에이전트 개발도구 : AgentBuilder의 활용

4-1. 지능형 에이전트의 체계

지능형 에이전트의 배경은 심리적(Psychological) 에이전트이다. 인지개발-정신모델에 기반한 에이전트-에 대한 기본적 작업은 에이전트의 체계에 대한 동기이다. Shoham[12] 인지 에이전

트가 여러 가지 정신적 요소를 구성하는 Beliefs, Commitments, Capabilities, 그리고 Commitment Rules가 있는데, 이들 정신상태를 인지적 에이전트로 소유한다는 것을 Shoham이 이론화했다. 또한 에이전트 프로그램 저술에서 에이전트프로그래밍 언어(APL, AGENT-0)를 정의했다.

Shoham의 작업을 보면, 그는 정신상태의 두 가지 주요 양상을 Belief와 Commitment로 기술했는데, Commitment는 목표를 추구하는 의사결정보다는 목표를 행하는 의사결정을 다루고 있고, Shoham의 'AGENT-0'는 두 개의 다른 행위타입을 정의했는데, 그것은 개별행위와 커뮤니케이션행위이다. 개별행위는 초기의 에이전트방법으로 에이전트의 환경에 영향을 미치는 일을 성취하는데 있으며, 예로 데이터베이스에이전트는 데이터베이스에서 SQL질의를 수행하는데 개별적 행위를 포함한다. 그리고 커뮤니케이션행위는 다른 에이전트와 메시지를 교환하는 메커니즘이다. 즉 이는 앞에서 언급했던 단독 에이전트와 다중 에이전트의 설명과 비슷한 내용이다.

에이전트의 행태가 복잡하게 수행될 때, 그것은 관계적으로 단순한 운영체제를 사용해 제공된다. 에이전트는 메시지를 주고받으며 개별행위를 수행하며, 그 자신의 정신모델을 업데이트할 수 있다. 에이전트의 행태는 프로그램으로써 관리되는데, Shoham의 AGENT-0 프로그램은 초기 Beliefs, 시초 Commitments, 에이전트의 Capabilities, 그리고 Commitment Rules로 구성돼 있다. 시초의 Beliefs는 에이전트의 시작점에서 정신상태를 나타낸다. 초기 Commitments는 에이전트가 미래의 어떤 특정시기까지 적용할 수 없을 때 발생한다. Capabilities는 에이전트가 에이전트의 활동시간에 대한 활동에 대해 수행할 수 있고 고정된 행위를 정의한다. 그리고 Commitment Rules는 모든 상황에서 수행된 행위와 에이전트의 정신적 변화를 결정한다.

Shoham의 AGENT-0는 Thomas의 PLACA(PLAnning Communication Agents)로 확장되었고, Reticular에서 더 추가해 에이전트 프로그래밍 언어(APL)를 확장했다. 이 객체지향언어를 RADL(Reticular Agent Definition Language)라 부르며, AgentBuilder는 Reticular의 Run-Time 시스템에서 수행되는 RADL프로그램을 생성하는 그래픽 툴을 제공한다. 그리고 이 툴에서는 KQML (Knowledge Query and Manipulation Language)를 사용하는데, 이는 분산된 어플리케이션 상에서 지능형 에이전트사이의 정보처리의 상호운영을 지원하기 위해 의도된 높은 수준의 언어이다. KQML은 에이전트사이에서 실행되어(Run-Time) 지식공유를 지원하기 위해 메시지 포맷과 메시지핸들링 프로토콜을 인정하고, 지능형 시스템과 커뮤니케이션하기 위해 어플리케이션 프로그램을 인정하는 언어이다. 또한 협력적으로 문제해결 시에 사용된 다중지능형 시스템들 사이에서 공유된 지식을 사용할 수 있다. KQML메시지는 'Performative(메시지의 집합, 선택적 논쟁의 집합)'로 구성되어 있는데, 이는 원격에이전트에서 가상지식베이스(Virtual Knowledge Base)를 시험하고 변화하는데 사용된 단언과 질의를 명세화 한다. AgentBuilder는 KQML에 대해 명세화된 모든 Performative지원을 제공한다.

4-2. 에이전트 개발 프로세스

(1) AgentBuilder의 특징

에이전트의 정신모델을 명세화하기 위해서는 초기의 Beliefs, 초기의 Commitments, 에이전트의 Capabilities, 그리고 Commitment Rules의 정의를 요한다. 여기서 지능형 에이전트 구축의 핵심은 행위 규칙을 명세화 하는 것으로 효율적 메커니즘과 정신모델의 다른 구성요소를 가지고 있다. 이런 것들을 수행하는데 있어, AgentBuilder(www.agentbuilder.com)는 쉽고 빠르게

에이전트의 집합을 정의하고 그들의 정신모델과 행위를 명세화 하는 그래픽인터페이스를 제공한다.

지능형 소프트웨어에이전트를 개발하는 것은 소프트웨어개발자가 시스템 분석 및 설계 상에서 분석, 설계, 구현, 테스트, 디버깅, 통합, 유지보수의 전통적 단계(SDLC)를 수행하는 다른 소프트웨어개발행동과 유사하다. 그리고 많은 면에서 소프트웨어에이전트의 개발은 객체지향소프트웨어개발과 유사하다.

객체지향 프로그래밍(OOP) 기술은 그 객체들 사이에서 객체의 이해를 인식하고, 여러 가지 커뮤니케이션을 명세화 하는 것이 있고, 다른 것으로 지능형 소프트웨어 에이전트 프로그램(ISAP) (때때로 에이전트 지향프로그램(AOP))라고 하는 것이 있다. 그리고 이들은 여러 가지 에이전트의 역할과 기능을 인식하는 것과 각 에이전트의 행위를 명세화하는 것으로 구성되며, 이 둘(OOP와 AOP)은 유사하다. 그리고 에이전트프로그래밍언어(APL)는 어떤 주어진 상황에 대해 에이전트의 행위를 명세화 하는데 사용된 높은 수준의 언어이다.

에이전트 엔진, 다시 말해 에이전트 수행환경은 에이전트프로그램을 수행하는데 요구되는 것이다. 이 엔진은 매우 광범위한 플랫폼 상에서 수행할 수 있고, 높은 성과를 제공하고, 복잡한 에이전트기반 어플리케이션의 생성을 지원해야 하는데, AgentBuilder는 이를 용이하게 하는 툴을 제공한다. 그 툴을 보면, 먼저 툴을 사용해 지능형 에이전트를 구축하는 프로세스로서 프로젝트 조직화와 통제를 하고, 문제영역을 분석하여, 에이전트 체계의 정의(상호작용 Protocols를 명세화, 에이전트체계정의, 에이전트 시험운영)를 하고, 또 에이전트 행태 명세화를 한다. 그리고 에이전트 응용프로그램생성 실행을 수행할 수 있는 에이전트들을 생성, 에이전트 & 에이전트 시를 검토 및 디버깅하는 순서로 구성되어있다.

(2) 에이전트 수행 프로세스

초기의 정신모델 상에서는 초기 Beliefs, 초기 Commitments, 에이전트의 Capabilities, 그리고 Commitment Rules로 기반으로 에이전트에 대한 정신모델을 정의하고, 이를 기초로 에이전트를 다음과 같이 수행한다. 그 사이클은 다음과 같다.

- ① 새로운 메시지처리 : 송신자를 인식하고 송신자의 출처가 확실한지 결정하는 것이다. 즉 그 메시지는 정신모델의 부분으로써 구성된다.
- ② 규칙이 현재상태에 적용할 수 있는지 결정 : 규칙이 현재상황과 조화되는가를 결정하는 것으로, 패턴매칭은 규칙을 만족하도록 결정하는 행태적 규칙 안에서 조건적 패턴으로 정신모델의 요소를 비교한다. 이 규칙은 모든 조건이 만족되었을 때 표시한다. 즉 규칙은 수행에서 에이전트의 아젠다로 놓여진다. 규칙의 수행은 개별적이며 커뮤니케이션적 행위를 수행하고 정신적 변화를 만드는데 그 목적이 있다. 여기서 모든 행위는 순서적으로 수행한다.
- ③ 규칙으로써 명세화된 행위를 수행 : 에이전트의 정신모델은 행위의 규칙수행에서 명세화된 정신적 요소를 추가하고 제거함으로써 업데이트 즉 그 규칙과 상응하여 정신모델을 업데이트된다.
- ④ 계획 단계 : 이는 에이전트의 계획을 개발하는 것을 요한다. 계획은 에이전트에 접촉된 기획모델으로써 수행되고 이 기획모델은 에이전트의 태도으로써 명세화된 목표를 만족시키는 계획으로 개발해야 한다는 것이다.

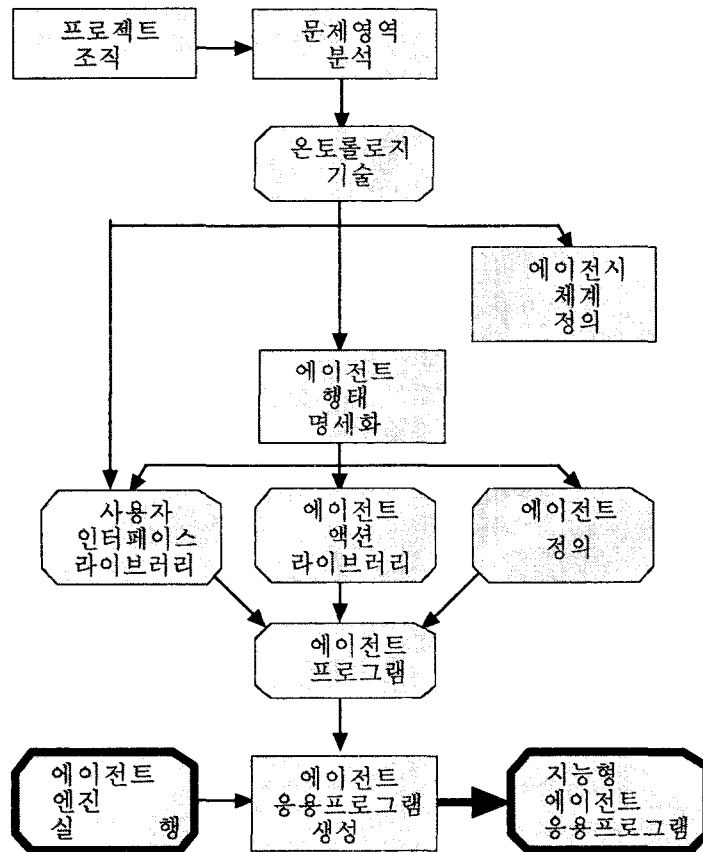
그리고 Intention, Beliefs, Commitments, Capabilities 각각은 에이전트 수행 사이클에 영향을 미치며 서로 교호작용을 한다.

에이전트의 전반적 환경은 다시 Message를 입수하여 처음단계로 Feedback된다. 마지막으로 시간체크기능이 있는데 이는 사이클에서 여러 단계를 수행하는데 적용되어 그들 시간을 감시하는데 사용된다.

(3) 에이전트 구현을 위한 프로세스

① Project 조직화와 통제

AgentBuilder 툴킷은 개발자에게 그들의 프로젝트를 조직하고, 특별한 에이전트와 에이전시를 수집하여 연계하게 하는데, 개발자는 관계된 프로젝트에서 미리 개발된 에이전트를 재사용할 수 있다. 그리고 특별한 영역에 대한 분석을 통해서 획득된 지식영역을 재 사용할 수 있다. 예로 email 에이전트를 구축한 개발자는 email에 대한 온톨로지를 개발할 수 있고, 이는 스팸-여과(Spam-Filtering) 에이전트의 개발을 요하는 새로운 프로젝트에 email 온톨로지를 재사용할 수 있다.



<그림-4> 지능형 에이전트 구현

이 툴킷으로, 프로젝트매니저에서 단순한 SellerBuyer 에이전트 구현을 위한 Project는, Seller 와 Buyer에 대한 트리(tree)구조 안에 파일을 생성한다. 이것은 나중에 다른 프로젝트를 생성할 때 쉽게 재사용 할 수 있다.

② 문제영역분석

개발자는 에이전트기능을 이해하고, 그의 에이전트요구와 에이전트기반 솔루션을 수행하기 위해 문제영역의 분석을 수행할 필요가 있다. AgentBuilder는 영역을 분석하고 구조화하고 그 영역에 관한 정보를 코드화하는 툴을 제공한다. 문제영역은 개념적 매핑 툴과 개념 그래픽 툴 그리고 객체지향모델링 툴을 사용해 문제영역을 용이하게 한다. 객체모델은 영역에서의 모든 객체와 그들을 수행할 수 있는 운영조직을 명세화한다.

영역분석의 또 다른 부분은 특별한 영역에 대한 온톨로지이다. 이 온톨로지는 문제영역을 정형적으로 기술(記述)하는데 커뮤니케이션시 충분히 이해할 수 있게 개념정의를 한다. 그리고 문제영역을 기술하기 위해 사용된 기호와 표현으로 방법을 제공한다. 한 에이전트가 또 다른 에이전트로 메시지의 의미를 적당하게 이해시키는 동안, 양쪽 에이전트는 그 메시지에서 사용된 같은 의미의 기호로 되어야 한다. 그 온톨로지는 문제영역에 대해 잘 이해할 수 있는 의미로 기호를 그린다.

이 단계에서는 문제영역이 분석되고 나면, 온톨로지 개념이 사용되어 온톨로지를 사용해 Seller와 Buyer에 대한 개념을 그림으로 표현한다. 즉 개념 모델러로 웹상에서 거래가 이루어지는 상황을 간략히 그림으로 표현하는 것이다. 그리고 오브젝트 모델러를 생성한다.

③ Agency체계의 정의

문제영역을 완전하게 한 후, 소프트웨어 개발자는 하나 또는 그 이상의 지능형 에이전트로써 수행할 수 있는 기능으로 문제를 분석한다. 개발자는 각 에이전트와 전체 문제를 해결할 때의 역할을 식별해야 한다. 그리고 에이전트 골격을 생성하고 다른 에이전트와 커뮤니케이션하여 그 에이전트의 기본적 특징을 정의 할 수 있다.

에이전트와 그들의 역할을 식별한 후에, 에이전트 개발자는 에이전트간 커뮤니케이션 프로토콜을 정의한다. 그리고 프로토콜 에디터는 개발자가 에이전트사이에서 요구되는 메시지와 웨이크헨딩을 명확히 하는데 쉽게 편집할 수 있는 툴을 제공한다. 그리고 나서 에이전트를 시험 운영한다.

이 단계에서는 문제영역에서 인식된 것들을 해결하기 위해, 에이전트의 역할을 명확히 하는 그래픽을 제공한다. 즉 Seller와 Buyer의 관계를 그래픽화 한다. 여기서 아이콘화된 그림으로 판매자의 역할과 구매자의 역할을 나타내 준다.

④ Agent 행태 명세화

에이전시가 정의 된 후에, 개발자는 각 에이전트의 행태로 명확히 한다. 에이전트개발은 에이전트 행태를 정의하는 프로세스이다. AgentBuilder는 행동규칙(Behavioral Rules), 최초신념(Initial Beliefs), Commitments, 의도(Intentions), 그리고 에이전트 능력(Capabilities)을 명확히 하는 툴을 제공한다. AgentBuilder 툴킷은 사용자 인터페이스 라이브러리(UI)와 에이전트 액션 라이브러리(AAL)의 생성을 지원하고, 에이전트 정의 파일을 생성한다. UI는 에이전트의 사용자 인터페이스를 구축하는데 사용되고 비록 많은 에이전트가 사용자와 함께 어떤 커뮤니케이션을 요구하지 않을 수 있지만, 다른 에이전트는 사용자로부터 정보를 필요로 하고 에이전트프로세싱의 상태에 관한 피드백 정보와 함께 사용자에게 제공될 것이다.

개발자는 AAL로 에이전트의 행위를 명세화한다. 각 에이전트의 행위에서 개발자는 다음 단계로 수행할 것이다. 첫째 행위이름과 변수리스트를 정의한다. 두 번째 객체모델에서 정의된 객체로부터의 방법과 함께 행위를 연계한다. 셋째 기존 클래스 라이브러리를 얻거나 행위를 구현할 수 있는 자바 클래스를 개발한다. 마지막으로 에이전트 행위 라이브러리에서 이들 클래스를 저장한다.

에이전트 정의 파일은 에이전트의 초기 정신모델과 행태의 상세한 명세서를 포함한다. 이 파일은 에이전트 행태를 완전히 명세화하기 위해 UIL와 AAL과 함께 사용한다. 이 세 가지 구성요소는 에이전트 프로그램으로 이루어진다.

이 단계는 에이전트가 수행되기 전에 에이전트에 대한 전반적인 상황을 나타내 주는데 액션을 통해 Seller와 Buyer의 행위를 명세화하고, Commitments로 실제 제품을 구매하려는 A와 판매하려는 B가 특정시간에 서로의 요구를 주고받는 행위를 명세화한다. 규칙편집기로 A가 하는 역할과 B가 하는 역할에서 여러 가지 규칙속성을 결정한다.

⑤ 에이전트 응용프로그램생성실행을 수행할 수 있는 에이전트들을 생성

에이전트구축 프로세스에서 마지막 단계는 Reticulars Run-Time 에이전트 엔진으로 에이전트 프로그램을 로딩하여 구성한다. 에이전트엔진은 에이전트 프로그램을 번역하고, 사용자 인터페이스와 AAL에서 명세화된 행위를 수행하는 성과수행 메카니즘이다. 이런 Reticulars 에이전트는 에이전트 프로그램과 Run-Time 에이전트로 구성되어 있다.

SellerBuyer에이전트프로그램이 제대로 구성되었으면, Run-Time엔진으로 에이전트어플리케이션을 생성하여 에이전트어플리케이션을 구축한다.

⑥ 에이전트와 에이전시 검토 및 디버깅

AgentBuilder는 소프트웨어 개발프로세스의 모든 상황을 지원하는 툴을 제공하고, 에이전트와 에이전시의 디버깅, 테스트, 통합을 지원하는 디버깅 환경을 제공한다.

AgentBuilder 디버거는 개발자에게 에이전트의 정신모델 시험을 인정하고, 개발자는 또한 에이전트의 운영 사이클 대 사이클을 통한 단계이며, 에이전트 수행자로서 정신모델을 시험한다.

지금까지 AgentBuilder로 에이전트를 구현하는 프로세스를 보았는데, 실제상에서는 보다 복잡하고 문제영역을 인식 할 수 있는 수준과 아울러 프로그래밍에 대한 기초를 요구하게 된다.

5. 결론 및 향후 연구과제

이상으로 본 논문에서는 전자상거래 프로세스를 개인 대 기업, 즉 구매자와 판매자간의 프로세스를 개념모델과 함께 AgentBuilder 툴 킷 상에서의 에이전트 프로세스를 구현에 대해 살펴 보았다. 그러나 현재 에이전트가 갖는 문제점은 고객의 입장에서는 새로운 매체에 대해 고객의 요구사항을 충분히 설명하지 못하고, 고객자신의 사생활 침해우려가 있기에 고객 개인의 정보 유출에 대한 불안감을 해소해야 한다. 판매자의 입장에서는 제품에 대한 충분한 표현이 요구되고 거래의 내용이 오용되면 경쟁사에 대해 경쟁력을 상실할 수도 있다. 그리고 기술적으로 현재 에이전트가 구현되어 운영되는 수준은 정보를 탐색하는데 있어 텍스트를 위주로 정보를 검색하고 있는데, 만약 이미지를 검색할 수 있는 에이전트가 개발되어 운영된다면, 보다 정확한