

전자상거래에서의 구매자 에이전트의 메모리 갱신 전략

김재욱*·이재규*

Memory Update Strategies for the Buyer Agent in Electronic Commerce

Jae Wook Kim*·Jae Kyu Lee*

요 약

구매 의사 결정 과정에서 구매자는 상품 정보와 가격 정보를 필요로 한다. 웹상의 판매자 위주의 시장으로부터 필요한 정보를 수집하려면, 구매자는 매번 구매 때마다 판매자의 사이트를 검색해야 한다. 심지어 구매자가 거래하고 있는 판매자로부터 과거에 구매한 경험이 있는 아이템을 구매하는 경우에도 마찬가지이다. 현재의 전자상거래 환경에서는, 이렇게 기억을 활용하지 않고(memory-less) 검색에만 의존하는 구매자의 행태가 일반적이다. 이 현상은 조직의 구매 경험이 구성원들에게 공유되어야 하는 기업 구매자의 경우에 있어서는 대단히 낭비적인 요소로 작용될 것이다.

구매자 에이전트에게 기억 능력을 부여하려면, 구매자 에이전트와 판매자 에이전트가 갖고 있는 정보간의 일관성을 유지해 주어야 할 필요가 있다. 최소의 노력으로 이러한 일관성을 유지하려면, 구매자 에이전트와 판매자 에이전트간에 체계적인 갱신 프로토콜이 필요하다. 본 연구에서는 6가지의 프로토콜(Instantaneous Pull, Periodic Pull, Instantaneous Push, Periodic Push, Instantaneous Solicited Push, Periodic Solicited Push)을 비교하고, 에이전트의 아키텍처와 KQML/XML에 기반 한 메시지 구조를 제시한다. 이들 프로토콜의 상황에 따른 수행 성능 평가가 진행 중이며, 이러한 상황에 적용할 수 있는 에이전트의 프로토타입이 현재 개발 중이다.

주제어: 전자상거래, 메모리, 일관성, 갱신 프로토콜, 구매자 에이전트, 판매자 에이전트, KQML, XML

Abstract

Buyers need the product and price information in making their purchasing decisions. To collect the necessary information from the seller-oriented marketplaces on the Web, buyers have to search around the sellers' sites for every purchase even though the buyers have purchase experience of certain items from certain sellers. This behavior of memory-less and search-only buyers happens in most of the current electronic commerce platforms. This phenomenon can be a waste to business buyers, whose organizational memory can be shared by multiple employees.

To allow the memory capability to buyer's agents, we need to keep the consistency between the buyer's agents and seller's agents. To assure the consistency with a minimal communication effort, the buyer's agent and seller's agents need to adopt a systematic protocol. We compare six possible protocols (instantaneous pull, periodic pull, instantaneous push, periodic push, instantaneous solicited push, and periodic solicited push), and propose the architecture of agents and messages based on the KQML/XML. A contingent performance is under evaluation and a prototype system, which can behave adaptive to the contingency, is under development.

Key words: electronic commerce, memory, consistency, update protocol, buyer agent, seller agent, KQML, XML

* 한국과학기술원 테크노경영대학원,

Graduate School of Management, Korea Advanced Institute of Science and Technology

1. 서론

상거래에 있어서 구매자가 구매 의사 결정을 내릴 때에는 구매하고자 하는 상품에 관한 정보를 필요로 한다. 비록 그 정보가 구매자가 예전에 본 적이 있는 것이라 하더라도, 실제 구매 의사 결정을 위해서는 그 상품에 관한 정확한 정보를 필요로 한다. 즉, 구매자의 상품 정보에 관한 기억이 완벽하지 않기 때문에, 정확한 정보를 얻기 위해 다시 검색해야 하는 경우가 발생한다.

또한, 구매자가 과거에 검색하였던 정보를 정확히 기억하고 있다 하더라도, 그 정보가 구매의사 결정을 내리는 시점에서 최신의(up-to-date) 정보라는 보장이 없다. 요즘같이 상품 정보가 자주 변하는 상황에서, 구매자가 최신의 정보를 기억하고 있지 않다면, 그 정보는 가치가 없게 된다. 따라서, 원래의 상품 정보의 변화가 구매자의 메모리(memory)에 반영될 필요성이 있다. 그러나 기억해야 할 상품 수가 증가함에 따라, 이러한 기억 과정과 메모리 갱신 작업을 인간이 행하는 데는 한계가 있다. 따라서, 구매자를 대신해서 상품 정보를 기억해 주는 구매자 에이전트와 이 에이전트의 메모리의 일관성을 유지하기 위한 방안이 필요하다.

구매자 에이전트의 메모리의 일관성은, 판매자 에이전트의 협조 없이는 유지되기 어렵다. 판매자 에이전트와 구매자 에이전트 간의 협업을 위해, 본 연구에서는 구매자 에이전트의 구조 뿐 아니라, 판매자 에이전트의 구조도 함께 제시하고자 한다. 구매자와 판매자 에이전트는 메시지 기반의 의사소통이 가능한 구조를 갖고 있다. 이를 위해 에이전트간 통신언어(ACL)로서 KQML과 XML을 사용하였다. 또한 구매자 에이전트의 메모리의 일관성 유지를 위해 에이전트간에 오가는 메시지와 상황에 따른 메모리 갱신 프로토콜을 설계하였다. 이러한 상황적인 메모리 갱신 프로토콜의 수행성과를 측정하기 위해 구매자 에이전트에 대한 프로토타입 시스템을 개발하고 있다

2. 관련 연구

전자상거래에서의 에이전트로서 가장 대표적인 것이 비교쇼핑 에이전트이다. 비교 쇼핑 에이전트란, 구매자를 대신해 인터넷 상에서 상품을 찾아주고 이들의 가격이나 사양을 비교해 주는 에이전트를 말한다. 이러한 비교쇼핑 에이전트의 효시는 앤더슨 컨설팅에서 개발된 BargainFinder(CSTaR, 1995)이다. 이 에이전트는 구매자가 원하는 CD를 가장 싼 값에 판매하는 인터넷 상점을 찾아 준다. 이러한 부류의 에이전트로는 BargainBot, ShopBot, Jango(Jango, 1997) 등이 있다. 그러나 이 에이전트들

은 상품 정보에 대한 완벽한 해석 능력을 갖고 있지 않다. 이것은 판매자의 상품 정보가, HTML이라는 구조화되지 않은 언어로 기술되어 있기 때문이다. 보다 완벽한 에이전트가 되기 위해서는, 에이전트가 상품 사양이나 가격 정보를 인지할 수 있는 형태로 상품 정보가 게시될 필요가 있다.

UNIK-AGENT(Lee and Lee, 1998)는 제품 검색, 선택, 계약 체결 과정을, 지능형 에이전트를 도입하여, 최대한 자동화하는 것을 목표로 하고 있다. 여기서는, 판매자와 구매자 에이전트간의 의사소통을 자동화 하기 위하여, 계약 유형을 사전에 정의하였다. 정의된 계약 유형은 구매자와 판매자 간의 매매 계약 체결 과정의 프로토콜을 결정하게 된다. 이 연구에서 제시한 UNIK-AGENT라는 지능형 에이전트의 아키텍처는, 전문가 시스템에 의사소통 능력과 문제해결 과정 제어 능력을 부여한 형태이다. 에이전트간의 의사소통을 구현하기 위해, 에이전트 통신 언어 계층(agent communication language layer), 전자상거래 계층(electronic commerce layer), 상품 사양 계층(product specification layer)이라는 3단의 메시지계층을 제시하였다.

다음으로, 데이터베이스 분야의 중복된 데이터 갱신에 관한 연구를 살펴볼 필요가 있다. 종종 여러 가지 이유로 인해, 동일한 데이터를 중복해서 저장할 경우가 있다. 그러나 이러한 중복성은 매우 신중히 관리되어야 한다. 이러한 중복 데이터의 갱신을 처리하는 가장 일반적인 방법으로 "Primary Copy Scheme (Date, 1995),"이 있다. 이 방법에서는, 복사된 객체 중 하나가 1차 복사본이 되고 나머지는 2차 복사본이 된다. 어떠한 갱신 사항도 논리적으로는 1차 복사본이 갱신되면 완료된 것으로 간주한다. 그리고 난 후, 1차 복사본을 갖고 있는 사이트에서 나머지 2차 복사본의 갱신을 책임지는 방법이다. 이러한 중복 데이터 갱신 이론은 구매자 에이전트의 메모리 갱신 프로토콜을 설계하는데 이용될 수 있다.

3. 에이전트 기반 상거래

3.1. 상거래: 판매와 구매

실제 상거래에서 판매자는 가능한 많은 상품을 팔려 할 것이다. 이를 위해 판매자는 구매자의 목록을 유지하고, 제품 카탈로그도 발송하며, 신상품이 출시되거나 기존 상품의 사양이나 가격이 변하면 이를 통지해 주기도 한다. 반면, 구매자의 목표는 가장 좋은 사양의 상품을 가장 싼 가격에 사는 것이다. 이러한 목표를 달성하기 위해, 구매자는 자신이 관심 있는 상품 정보를 검색하고, 검색한 정보를 유지 관리하다가, 구매 의사결정 과정에 활용한다.

전자상거래에 있어서도 이러한 구매자와 판매자의 본성은 다를 것이 없다. 다만, 구매자가 검색하는 상품 정보의 근원이 디지털 자료이고, 판매자가 보내주는 상품 카탈로그가 전자 카탈로그라는

점이 다를 뿐이다.

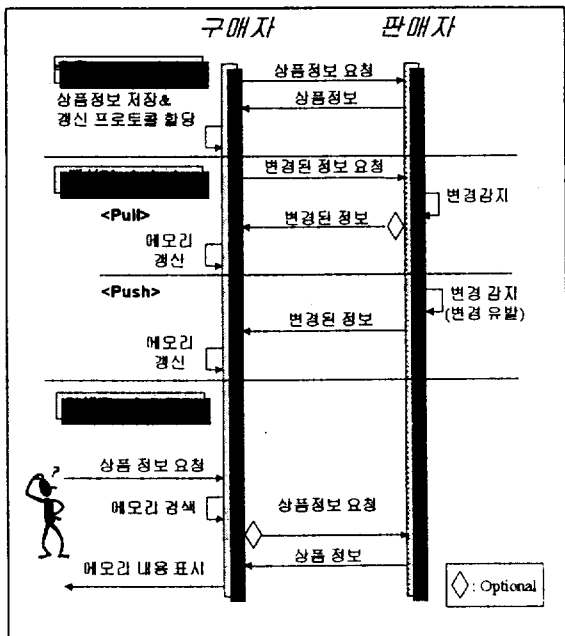
판매자는 자신이 판매하는 상품 정보를 데이터베이스 등에 저장하고 있다가, 이러한 정보를 기반으로 전자 카타로그를 생성하여 마케팅 활동을 전개할 것이다. 제품 카타로그에 포함 되는 주요 정보는 제품의 사양과 가격이다. 그런데 이 가격과 사양은 요즘같이 급변하는 시장 환경에서는 매우 유동적일 수 밖에 없다.

구매자에게 있어서는, 정확하고 풍부한 제품 정보 없이는 성공적인 구매 의사결정을 내리기 어렵다. 그러나, 정보량이 증가할수록, 더 많은 기억 능력이 필요하게 되는데 비해, 인간의 기억에는 한계가 있다. 구매자 에이전트는 이러한 한계를 해결해 줄 수 있다. 그러나, 구매자 에이전트의 메모리가 자주 갱신되지 않는다면 이 또한 별 소용 없는 낡은 정보가 되어버릴 것이다.

결국, 이러한 이유로, 원활한 전자상거래 활동을 위해, 판매자나 구매자 모두 지능형 에이전트가 필요한 것이다.

3.2. 메모리 축적과 일관성 유지

본 연구에서는, 전자상거래상의 구매자를 위한 에이전트 기반의 메모리 축적과 일관성 유지를 위한 프레임워크(Framework)를 제시하고자 한다. 연구의 주 관심은 구매자 에이전트의 메모리 갱신에 있지만, 메모리하고 있는 데이터가 상품 정보이기 때문에, 이 상품 정보와 구매자 에이전트의 메모리간의



일관성 유지를 위해서는 판매자 에이전트의 도움이 필요하다. 따라서 프레임워크에는 구매자 뿐만이 아니라 판매자 에이전트의 구조(architecture)도 포함되어 있다.

[그림 1] 메모리 축적, 갱신, 검색 시나리오

에이전트 메모리 활용 시나리오는 3 단계로 구성되어 있다: 메모리 축적, 갱신, 검색(Retrieval). [그림 1]은 에이전트간의 협업을 통한 메모리 축적, 갱신, 검색에 대한 전반적인 시나리오를 보여주고 있다.

메모리 축적 단계. 구매자가 웹을 검색하는 과정에서, 자신에게 필요한 상품 정보를 보게 되면, 이를 정확히 기억해 두기를 원할 것이다. 기억해 두고자 하는 메모리의 내용은 상품의 사양과 가격 정보가 될 것이다. 구매자 에이전트는 구매자를 대신해, 구매자가 원할 경우, 해당 상품 정보(사양과 가격)를 자신의 메모리에 저장하게 된다. 이때 저장된 정보의 일관성을 유지하기 위한 갱신 프로토콜도 함께 지정한다.

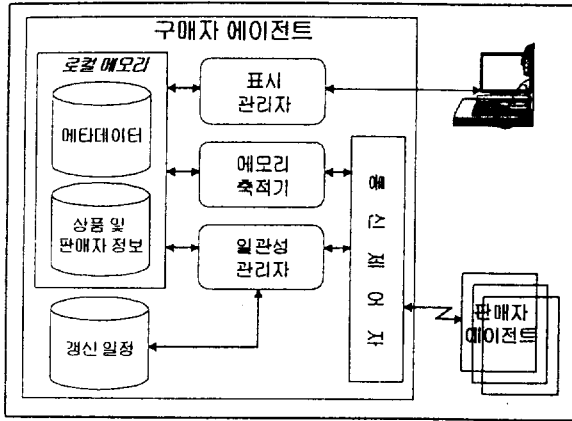
메모리 갱신 단계. 상품의 사양이나 가격이 변할 경우, 이러한 변경사항은 고객 에이전트의 메모리에 반영되어야 한다. 이러한 변경사항을 반영하는 방법은 변경 작업을 주도하는 주체가 누구냐에 따라 "Pull"과 "Push" 방식으로 나누어 볼 수 있다.

구매자 에이전트가 판매자 에이전트에게 변경된 정보를 요청하여 자신의 메모리를 갱신하는 방법이 Pull 방식이다. 반대로, 판매자 에이전트가 구매자 에이전트의 메모리 갱신을 주도(trigger)하는 것은 Push 방식이다. 이때는, 판매자 에이전트가 자신이 갖고 있는 상품 정보에 변화가 있을 때, 이를 구매자 에이전트에게 알려주어 구매자 에이전트가 자신의 메모리를 갱신하도록 해 준다. 메모리의 갱신에 관해서는 5절에서 상세히 살펴보도록 하자.

메모리 검색(Retrieval) 단계. 구매자가 의사 결정을 위해 상품 정보가 필요한 경우, 그 정보가 과거에 검색한 적이 있는 정보라면 구매자 에이전트는 자신의 메모리를 검색하여 그 결과를 구매자에게 제공할 것이다. 만약 필요한 정보가 메모리에 존재하지 않는다면, 일반적인 웹 검색 절차를 따르게 된다. 그러다가 구매자가 검색된 정보를 기억해 놓고 싶다고 판단하면, 메모리 축적 단계가 다시 시작되게 된다. 이후의 과정은 순환적으로 계속 된다.

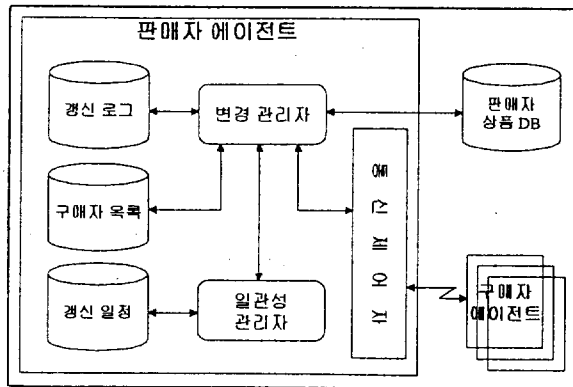
3.3. 구매자와 판매자 에이전트

구매자 에이전트 앞에서 제시한 프레임워크를 지원하기 위해서는, 구매자 에이전트는 [그림 2]와 같은 구조를 가져야 한다. 메모리 축적 단계에서 검색된 상품 정보를 저장하기 위한 공간이 에이전트의 로컬 메모리(Local Memory)이다. 로컬 메모리는 상품 정보, 판매자 정보, 메타데이터로 구성된다. 상품 정보는 상품 사양과 가격으로 구성된다. 관심 상품을 취급하는 판매자 정보 또한 로컬 메모리의 저장 대상이다. 메타데이터는 로컬메모리가 갖고있는 내용에 관한 정보를 저장하고 있다. 즉, 구매자 에이전트가 자신의 로컬 메모리에 어떤 상품에 관한 정보와 어떤 판매자에 관한 정보를 저장하고 있는지에 관한 정보가 메타데이터이다. 구매자 에이전트는 메모리의 일관성 유지를 위한 갱신 일정도 저장하고 관리하고 있다.



[그림 2] 구매자 에이전트의 구조

메모리 축적기는 로컬 메모리에 상품이나 판매자 정보를 추가해 주는 역할을 담당하고 있다. 구매자가 저장된 상품 정보나 판매자 정보를 필요로 할 때에는, 표시(presentation) 관리자가 메모리의 내용을 가져와 사용자에게 보여준다. 변경 일정에 따라, 로컬 메모리를 갱신해 주는 것은 일관성 관리자가 담당하고 있다. 또한, 에이전트간의 협업 과정에서 필요한 메시지의 수신 및 발송은 통신 제어자가 맡고 있다. 에이전트들 간의 의사소통은 에이전트통신언어 (Agent Communication Language: ACL)



를 사용한다. 자세한 사항은 5절에 있다.

판매자 에이전트 [그림 3]은 판매자 에이전트의 구조를 보여 주고 있다. 변경관리자는 판매자의 상품 데이터베이스내의 변경 사항을 감지하여 갱신 로그에 저장하는 역할을 담당한다. 갱신 로그는 데이터의 변경사항을 담고 있다. 일관성 관리자가 구매자 에이전트에게 보낼 변경 사항을 요청하면, 변경 관리자는 갱신 로그를 검색하여 요청된 데이터를 전달한다.

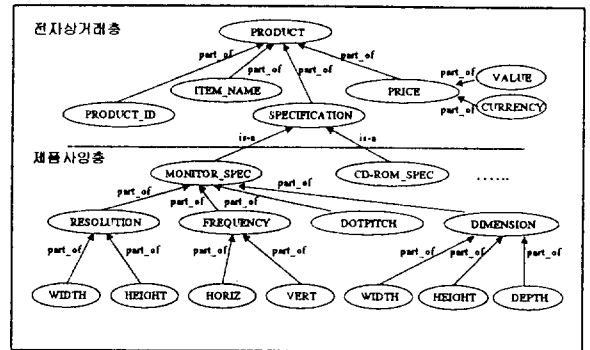
구매자 목록은 Push 갱신에서 필요로 하며, 에이전트간의 의사소통은 통신관리자가 담당한다.

[그림 3] 판매자 에이전트의 구조

4. 구매자 에이전트 메모리의 표현

4.1. 구매자 에이전트 메모리의 계층

구매자 에이전트의 메모리는 계층 구조로 구성되어 있다: 전자상거래층(Electronic Commerce Layer)과 제품사양층(Product Specification Layer). 이러한 계층 구조는 앞서 살펴본 UNIK-AGENT (Lee and Lee, 1998) 연구로부터 채택하였다. [그림 4]는



구매자 에이전트 메모리의 2 계층 구조를 보여 주고 있다.

[그림 4] 구매자 에이전트의 메모리 계층

“전자상거래층”은 상거래에서 상품을 기술하는 일반적인 속성들을 정의한다: 제품아이디 (PRODUCT_ID), 아이템명 (ITEM_NAME), 가격 (PRICE) 등. “제품사양층”은 아이템 레벨의 개별 제품의 속성으로 구성된다: 예를 들어 모니터 (MONITOR_SPEC)의 사양의 경우에는, 해상도 (RESOLUTION), 주사빈도(FREQUENCY), 도트피치 (DOTPITCH) 등의 속성으로 구성되어 있다. 그러나, 이러한 속성들은 제품 카테고리에 따라 다르며, 속성명의 동의어(Synonym)도 고려하여야 한다.

4.2. 구매자 에이전트 메모리의 관리

웹상에 정보를 게시하는 데 가장 널리 사용되는 방법이 HTML이다. 그러나 HTML은 Presentation 용 언어이기 때문에, 정보의 표시 형식을 제어하고 사람이 그것을 보는 데는 적합하지만, 의미론적인 부기(Semantic Markup)를 가하기는 어렵다. 그러나 에이전트 기반의 전자상거래에서, 각각의 항목이 의미하는 바가 무엇인지를 아는 것은 매우 중요하다 (예를 들어 “이 항목은 상품 가격이다,” “이 항목은 모니터의 해상도이다”). 에이전트의 메모리는 당연히 기계 해석이 가능하여야(Machine Interpretable) 한다. 본 연구에서는 에이전트의 메모리를 표현하는데 HTML이 아닌 XML을 채택하였다. XML (W3C-XML, 1998)은 에이전트에게 의미론적 부기 능력을 제공해 준다. XML 데이터는 XSL (W3C-XSL, 1999)을 이용하면, HTML 형식으로 브라우저 가능하다. XML과 XSL을 통하여 에이전트 메

모리 자체의 내용과 화면 표시를 분리할 수 있다.

5. 메모리 일관성 유지

5.1. 메모리 갱신 프로토콜

메모리 갱신 프로토콜의 분류. 갱신 프로토콜은 갱신 시점과 시작자(Initiator)라는 두 가지 기준에 의해 분류할 수 있다. 갱신 시점 기준으로는 즉각적 갱신(Instantaneous Update), 주기적 갱신(Periodic Update)으로 나눌 수 있다. Push 갱신과 Pull 갱신은 누가 갱신 작업을 시작했는가를 관점에서의 분류이다.

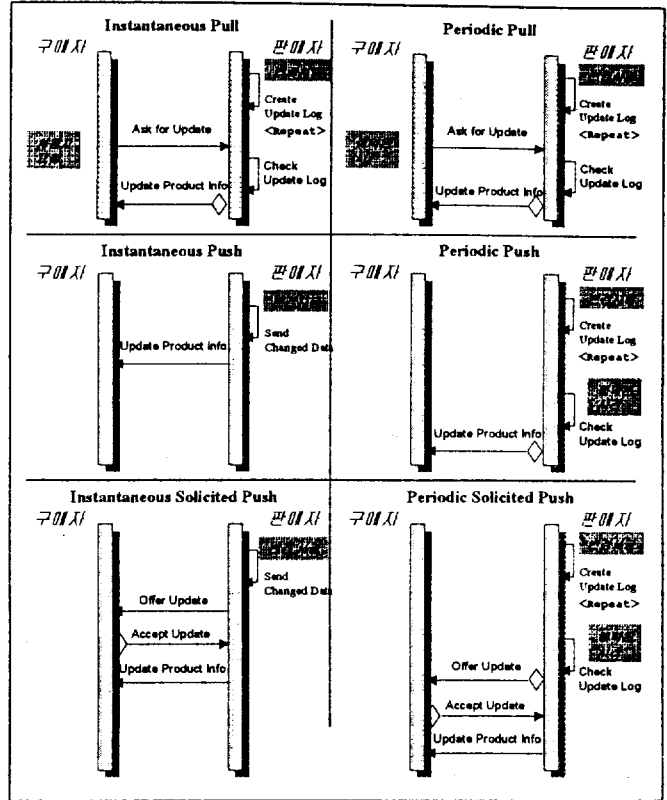
즉각적 갱신 프로토콜은 판매자 측의 데이터에 변경이 생긴 즉시 구매자 에이전트의 메모리에 반영하는 방식이다. 일관성 유지 측면에서는 이 프로토콜이 최선이라 할 수 있으나, 자칫 막대한 네트워크 부하를 유발할 가능성이 있다. 이 프로토콜은 제품 가격과 같이, 구매자의 의사 결정에 결정적인 영향을 미치는 데이터의 갱신에 적합할 것이다. 주기적 갱신 프로토콜 하에서는, 변경된 데이터의 갱신이 미리 정해진 시점까지 연기되는 프로토콜이다. 즉, 데이터의 갱신이 주기적으로 이루어지게 된다. 이 프로토콜은 필요 없는 갱신 노력을 줄일 수 있다는 장점이 있으나, 대기 시간이 오래 걸릴 수도 있다는 단점이 있다.

Pull 갱신 프로토콜은 구매자 에이전트가 판매자 에이전트에게 갱신할 데이터를 요구하는 방식이다. Push 갱신 프로토콜 하에서는, 판매자 에이전트가 변경된 데이터를 구매자 에이전트에게 전송해 준다. 이 프로토콜은 구매자의 편의성 측면에서는 우수하나, 과도한 네트워크 부하와 불필요한 갱신 노력 발생 가능성이 존재한다.

혼합 갱신 프로토콜. 이상의 갱신 프로토콜들을 조합 하면 다음의 <표 1>과 같은 혼합 갱신 프로토콜을 얻을 수 있다. Instantaneous Pull, Periodic Pull, Instantaneous Push, Periodic Push, Instantaneous Solicited Push, Periodic Solicited Push

<표 1> 혼합 갱신 프로토콜

	Instantaneous	Periodic
Pull	Instantaneous Pull	Periodic Pull
Push	Instantaneous Push	Periodic Push
Solicited Push	Instantaneous S. Push	Periodic S. Push



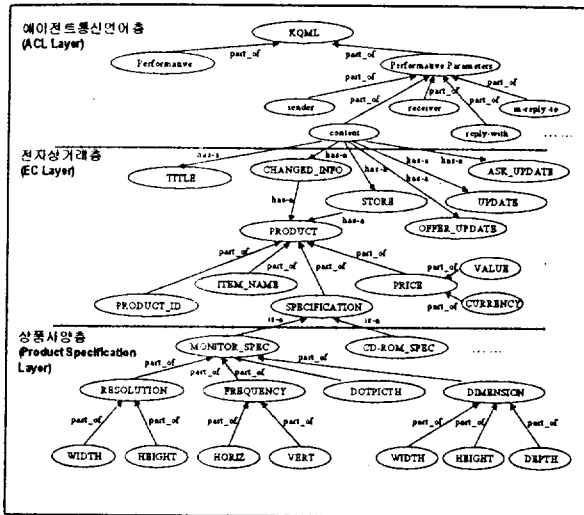
[그림 5] 혼합 갱신 프로토콜별 메시지 흐름

5.2. 메모리 축적과 일관성 유지를 위한 에이전트 통신언어

에이전트 통신 언어. 에이전트간의 의사 소통을 위해, 에이전트는 메시지를 생성하여 전송하고, 받은 메시지를 해석하는 일들을 한다. 이를 위해 보다 정형적이고 일반적인 메시지의 표현, 즉 에이전트 통신언어(Agent Communication Language; ACL)가 필요하게 되었다. ACL은 크게 외부 언어(Outer Language)와 내부 언어(Inner Language)(Genesereth and Ketchpel, 1994) 두 부분으로 구성된다. 본 연구에서는, Knowledge Query and Manipulation Language (KQML)를 외부 언어로, XML을 내부 언어로 채택하였다.

ACL은 본래 적용 영역에 무관하게, 일반적인 에이전트간의 의사소통을 위해 고안되었다. 따라서, ACL을 본 연구에 적용하기 위해서는, 내부 언어를 에이전트 기반의 메모리 갱신 작업을 보다 잘 표현할 수 있도록 세분화해 정의할 필요가 있다. 이를 위해, UNIK-AGENT (Lee and Lee, 1998) 연구에서와 같이, 3계층의 메시지 구조를 채택하였다. 3계층은 에이전트 통신 언어 계층(agent communication language layer), 전자상거래 계층(electronic commerce layer), 상품 사양 계층(product specification layer)이다.

메시지 계층. 구매자와 판매자 간의 메시지는 [그림 6]에서 보는 바와 같이 3 계층으로 구성되어 있다.



[그림 6] 메시지의 3 계층

가장 상위 층(에이전트 통신 언어 계층)은 KQML 퍼포머티브(Performative)와 퍼포머티브 인수(Performative Parameters) (Finin, et al., 1993)로 구성되어 있으며, 이 계층은 도메인과 무관하다.

전자상거래 계층은 KQML content에 메모리 갱신에 특화 된 인수를 추가한 것이다. 예를 들어, TITLE은 메시지의 이름을 나타내고, STORE는 메모리에 상품 정보를 저장하도록 지시하는 것을 나타낸다. ASK_UPDATE는 메모리의 갱신을 요청하는 것이다. CHANGED_INFO는 PRODUCT_ID, ITEM_NAME, PRICE, SPECIFICATION으로 구성된 상품 정보를 갖고 있다.

상품 사양 계층은 상품 아이템을 보다 상세히 정의한다. 이 계층은 에이전트 메모리에 축적되거나 갱신될 제품 사양으로 구성되어 있다.

메시지 예 [그림 7]은 메시지의 예로서, 구매자 에이전트의 메모리에 상품 정보를 저장하라는 내용이다.

```

TELL
:SENDER      S3
:RECEIVER    B1
:ONTOLOGY    Agent Learning and Consistence
:LANGUAGE    XML
:CONTENT
[<? XML VERSION="1.0"?>
<CONTENT>
<TITLE>STORE</TITLE>
<PRODUCT>
<MONITOR>
<PRODUCT_ID> LG17XL700P </PRODUCT_ID>
<ITEM_NAME> Monitor </ITEM_NAME>
<PRICE>
<VALUE>500</VALUE>
<CURRENCY>USD</CURRENCY>
</PRICE>
<SPECIFICATION>
<RESOLUTION>
<WIDTH UNIT="pixel"> 1600 </WIDTH>
<HEIGHT UNIT="pixel"> 1200 </HEIGHT>
</RESOLUTION>
<FREQUENCY>
<HORIZ UNIT="KHz"> 30-85 </HORIZ>
<VERT UNIT="KHz"> 50-120 </VERT>
</FREQUENCY>
<DOTPITCH UNIT="mm"> 0.26 </DOTPITCH>
<DIMENSION>
<WIDTH UNIT="mm"> 544 </WIDTH>
<HEIGHT UNIT="mm"> 530 </HEIGHT>
<DEPTH UNIT="mm"> 533 </DEPTH>
</DIMENSION>
</SPECIFICATION>
</MONITOR>
</PRODUCT>
</CONTENT>
]

```

[그림 7] 메시지 예

5.3. 갱신 프로토콜의 수행 성능평가

가상 우수한 수행 성능을 나타내는 메모리 갱신 프로토콜은 무엇일까? 이것은 변화되는 데이터의 특성에 따라 다를 수 있다. 따라서, 여기서는 상황에 따른 갱신 프로토콜의 수행 성과를 측정하고자 한다.

성능 평가 기준. 갱신 프로토콜의 수행 성능 평가를 위한 기준은 응답 시간과 네트워크 부하가 될 수 있다.

(1) 응답 시간: 대부분의 사용자는 컴퓨터 화면 앞에서 기다리는 시간에 매우 민감하다. 따라서, 메모리 검색 시의 반응 속도는 갱신 프로토콜의 수행 성과 측정의 중요한 기준이 된다.

(2) 네트워크 부하: 비록 어떤 메모리 갱신 프로토콜이 아무리 많은 장점을 갖고 있다 해도, 만약 이로 인해 네트워크에 엄청난 부하가 걸린다면, 이 프로토콜을 사용하기는 힘들 것이다.

이상의 기준으로 본 연구에서 제시한 메모리 갱신 프로토콜의 수행 성과 측정을 위해, 다음과 같은 측정 수단(Measure)이 필요하다.

응답 시간은, 사용자의 검색 요구로부터 결과 표시까지 소용된 시간을 말하며, 이는 Seconds Per Hit (SPH)로 측정할 수 있다.

• Min. Seconds Per Hit (Min. SPH)

- Max. Seconds Per Hit (Max. SPH)
- Average Seconds Per Hit (Avg. SPH)

네트워크의 부하 정도는, 메모리 갱신을 위해 네트워크 상에 전송되는 패킷의 분량으로 측정 가능하다.

- Number of Packets Sent and Received

프로토타이핑과 수행 성능 평가 이러한 프로토콜들의 수행 성능 평가는 일률적으로 어느 프로토콜이 좋다고 말하기 보다는, 어떤 상황에서는 어느 프로토콜이 더 좋다는 식의 상황적인 평가가 이루어져야 할 것이다. 이를 위해 각 프로토콜의 상황에 따른 수행 성능 평가가 진행 중이며, 이러한 상황에 적용할 수 있는 에이전트의 프로토타입이 현재 개발 중이다.

6. 결론

본 연구에서는 다음과 같은 작업을 수행하였다:

첫째, 전자상거래에 있어서 구매자 에이전트의 메모리 갱신을 위한 프레임워크를 제시하였다. 여기서 제안한 프레임워크는 판매자와 구매자 에이전트의 구조와 KQML/XML을 채택한 ACL로 작성된 메시지 구조를 포함하고 있다.

다음으로, 구매자 에이전트의 메모리 갱신을 위한 상황적 프로토콜을 설계하였다.

마지막으로, 설계된 메모리 갱신 프로토콜의 수행 성과 측정을 위한 평가 기준과 측정 수단을 정의하였고, 해당 프로토타입 시스템을 개발 중이다.

참고 문헌

[CSTaR, 1995] Andersen Consulting Center for Strategic Technology Research (CSTaR). 1995. BargainFinder. <http://bf.cstar.ac.com/bf/>.

[Jango, 1997] Jango. 1997. <http://www.botspot.com/pcai/index.html>

[Lee and Lee, 1998] Jae Kyu Lee and Woonkyu Lee. 1998. An Intelligent Agent-Based Competitive Contract Process: UNIK-AGENT. International Journal of Intelligent Systems in Accounting, Finance & Management 7:91-105.

[Date, 1995] C. J. Date. 6th eds. 1995. An Introduction to Database Systems. Addison-Wesley.

[Genesereth and Ketchpel, 1994] Genesereth, M. and Ketchpel, P. 1994. Software agents. Communications of the ACM, 37(7).

[Finin et. al., 1993] Finin, T.; Weber, J.; Wiederhold, G.; Generaeth, M.; Fritson, R.; McGuire, J.; Shapiro, S.; and

Beck, C. 1993. Specification of the KQML agent-communication language plus example agent policies and architecture. Draft, the DARPA Knowledge Sharing Initiative, External Interface Working Group.

[W3C-XML, 1998] Extensible Markup Language (XML) 1.0. W3C Recommendation. February 10, 1998. <http://www.w3.org/XML/>

[W3C-XSL, 1999] W3C Working Draft. April 21, 1999. <http://www.w3.org/TR/WD-xsl/>