

인터넷 상품정보 추출을 통한 M-commerce 콘텐츠 제공자 시스템의 설계 및 구현

하 상 호*

요 약

최근에 WCDMA, LTE 등 이동통신망의 기술 발달과 모바일 폰의 성능 향상과 보급 확대로 모바일 폰을 이용한 M-commerce가 급속도로 확산되고 있다. 모바일 쇼핑몰은 보통 인터넷 쇼핑몰과는 다른 별도의 사이트로 구축되어 운용되고 있다. 이는 별도 사이트 구축에 따른 비용 추가는 물론, 유·무선 환경간의 콘텐츠 비연동으로 인하여 사이트 운영의 비효율성을 초래한다. 본 논문에서는 인터넷 쇼핑몰로부터 사용자에게 관심 있는 상품 항목을 추출하여 모바일 쇼핑몰 콘텐츠를 구성하는 M-commerce 콘텐츠 제공자 시스템을 개발하고, 인터넷 쇼핑몰에 적용하고, 평가한다. 평가 결과는 사용자의 모바일 폰에 전송되는 데이터가 일반 웹 브라우저를 사용한 경우와 비교해서 크게 감소된다는 것을 보여준다.

키워드 : 인터넷 쇼핑몰, 무선 상거래, 모바일 폰, 콘텐츠 제공자

Design and Implementation of a M-commerce Content Provider System by Extracting the Internet Product Information

Sangho Ha*

ABSTRACT

M-commerce services are rapidly emerging with recent advances in mobile devices and wireless communications. Internet Shopping mall managers provide mobile phone users with their mobile shopping malls for m-commerce. However, these mobile shopping malls usually have their own sites which are not compatible with their Internet shopping malls, thus resulting in inefficiency to manage the two kinds of shopping malls. In this paper, we develop a M-commerce content provider system to effectively bring Internet shopping malls to mobile phones by extracting items only concerned to users from product documents. We then apply and evaluate the system over a specific internet shopping mall site. The results show that the system reduces considerably the amount of data transferred to user's mobile phones compared with using general web browsers.

Keywords : Internet Shopping Malls, M-commerce, Mobile Phones, Content Provider

1. 서 론

최근에 WCDMA, LTE 등 이동통신망의 속도가 크게 향상되고, 스마트 폰이나 태블릿 PC 등의 모바일 기기 보급이 확산되고, 모바일 결제 시스템 관련 기술이 향상됨에 따라서 모바일 폰을 이용한 M-commerce가 급속도로 확산되고 있다. 실제로, 2011년 현재 M-commerce에 의한 전 세계 거래가 2,400억 달러이며, 2015년에는 약 3배 증가한 6,700억

달러에 이를 것으로 전망되고 있다[1]. 국내에서도 M-commerce의 매출액이 2011년 6,274억 원이며, 이는 2015년에는 2조 6,500억 원의 수준으로 급격하게 증가할 것으로 전망되고 있다[2].

M-commerce의 사례를 살펴보면, 해외에서는 NTT DoCoMo의 i-Mode 포탈[3,4], Webraska의 스마트존 플랫폼 [5], Vodafone의 m-payment[6] 등을 들 수 있고, 국내의 경우에는 SKT, KT 중심으로 M-commerce 서비스가 부분적으로 제공되고 있다. SKT[7]의 MONETA, NeMo와 KTF [8]의 nPayMagic, 쇼터치는 휴대폰 이용 지불을 위한 모바일 뱅킹 서비스를 제공하고 있으며, SKT, KTF 모두 휴대폰 상에서 쇼핑몰 검색이 가능하도록 M-Marketplace를 제공한다. 또한 최근에는 스마트폰을 이용한 구글의 모바일

* 이 논문은 순천향대학교 교수 연구년제에 의하여 연구하였음.

† 종신회원: 순천향대학교 컴퓨터공학과 교수

논문접수: 2012년 5월 3일

수정일: 1차 2012년 6월 14일

심사완료: 2012년 6월 20일

* Corresponding Author: Sangho Ha(hsh@sch.ac.kr)

Wallet[9], 페이스북의 F-Commerce[10], SKT의 모바일 11번가[11], T스마트윌렛[12] 등의 스마트폰 전용 애플리케이션이 나와서 M-commerce의 시장을 급속도로 확산시키고 있다.

일반적으로 모바일 쇼핑물은 인터넷 쇼핑물과는 다른 별도의 사이트로 구축된다. 이는 휴대폰의 작은 화면, 처리 능력 제한성, 인터넷 비해 현저히 느린 이동 통신망의 속도등을 고려하여 휴대폰 기기에 최적의 콘텐츠를 제공하기 위함이다. 이러한 상황은 스마트폰의 보급과 4G의 통신기술의 발달로 인하여 크게 나아지고 있지만, 여전히 쇼핑물 11번가와 같이 쇼핑물에 대한 유·무선 콘텐츠가 별도의 사이트를 통해서 제공되고 있다. 이는 별도 사이트 구축에 따른 비용 추가는 물론, 유·무선 환경간의 콘텐츠 비연동으로 인하여 사이트 운영의 비효율성이 초래된다.

본 논문에서는 인터넷 쇼핑물과 연동하여 M-commerce를 위한 콘텐츠를 효과적으로 제공할 수 있는 시스템을 개발하였다. 인터넷 쇼핑물로부터 상품을 검색하는 과정을 살펴보면, 먼저 인터넷 쇼핑물 사이트에 접근하고, 이 사이트에서 쇼핑물이 제공하는 검색 엔진을 통해서 관심 있는 상품을 검색한다. 다음에, 검색된 상품 문서에서 가격, 제조사와 같은 관심 있는 상품 항목을 살펴보게 된다. 이러한 검색 과정을 반영하여, 시스템은 인터넷 쇼핑물 검색에 필요한 제반 정보로, 쇼핑물 사이트 주소, 쇼핑물 검색 엔진 주소, 상품 문서의 주소 패턴, 관심 있는 상품 항목 등을 래퍼(wrapper)에 표현하고, 나중에 이 래퍼를 이용하여 사용자에게 관심 있는 상품 항목만을 직접 검색하는 기능을 제공한다.

여기서 중요한 것은 상품 문서에 대한 검색이 아니고, 문서에 포함된 상품 항목에 대한 검색이라는 것이다. 상품 문서에는 보통 사용자에게 관심 없는 다른 상세 사항을 다량 포함한다. 사용자에게 관심 없는 사항을 제거하고 반드시 필요한 상품 항목만을 제공함으로써 모바일 폰 상에서 사용자의 상품 검색 편의성을 향상시키고, 불필요한 데이터 전송에 따른 통신 비용을 절감할 수 있다.

상품 문서에 포함된 특정 항목에 어떻게 직접 접근할 수 있는가? HTML 문서는 DOM(Document Object Model) 트리 구조로 표현되기 때문에 항목의 경로는 트리의 루트로부터의 유일한 경로를 갖게 된다. 따라서 상품 항목에 대한 문서상의 경로를 구하여 래퍼에 포함시키면 된다. 논문에서는 [13]의 웹 항목 추출 시스템의 관련 컴포넌트를 이용하여 상품 항목의 문서상의 경로를 산출하여 래퍼에 포함시키고, 나중에 래퍼에 포함된 항목 경로를 이용하여 문서상의 항목에 직접 접근한다.

한 가지 더 눈여겨볼 사항은, 상품 문서의 '주소'가 아니라 '주소 패턴'이 래퍼에 포함된다는 것이다. 이는 한 인터넷 쇼핑물에서 동일 상품 분류에 속한 상품 문서의 구조는 동일하고, 이러한 문서들에 대한 URL 주소 패턴이 일치한다는 사실을 반영한다. 가령, 상품 문서의 주소 패턴이 한 상품 분류 '컴퓨터'에 대한 것이라면, 특정 컴퓨터의 상품 문서에 포함된 특정 상품 항목 '가격'의 경로는 컴퓨터 분류에 속한 다른 모든 문서에 포함된 '가격'의 항목의 경로와 동일하다. 따라

서 래퍼에 포함된 상품 항목 경로는 동일 상품 분류에 속한 모든 상품 문서의 항목들을 검색하는데 사용된다.

관련 연구로, [14]에서는 인터넷 쇼핑물의 상품정보를 모바일 폰에 효과적으로 제공하는 것에 관한 연구를 수행하였다. 그러나 이 연구는 인터넷 쇼핑물 문서가 HTML 문서가 아닌 XML 문서임을 가정하고 있다. 즉, 인터넷 쇼핑물의 상품 콘텐츠가 E-commerce를 위한 상품표현모델[15]에 기반하여 XML 문서로 작성되었다고 가정한다. [14]에서는 이러한 XML 문서를 XSLT[16]를 이용하여 모바일 폰을 위한 상품 표현모델, 3MP(Merchant Model for Mobile Phones)[14]의 XML 문서로 변환하여 모바일 단말기에 전달한다. 반면에, 본 논문에서는 대부분의 인터넷 쇼핑물의 콘텐츠를 표현하는 HTML 문서를 고려한다. 이러한 문서로부터 사용자에게 관심 있는 상품 항목들을 추출하고, 추출된 항목들을 3MP 모델의 XML 문서로 구성하여 모바일 단말기에 전달한다는 점에서 [14]와는 그 접근 방법이 다르다.

또한, [17, 18, 19]에서는 웹 문서로부터 정보를 추출하는 연구가 진행되었다. 그러나 이러한 연구 모두 [13]과 마찬가지로 일반적인 웹 정보 추출에만 그 연구가 한정되며, 인터넷 쇼핑물 사이트로부터의 상품항목 추출에 기반하여 E-commerce와 연동되는 M-commerce에 대해서 고려하지 않는다. [20, 21, 22]에서는 웹 페이지를 화면이 작은 모바일 단말기에 효과적으로 디스플레이하도록 적용시키는 연구가 진행되었다. 이러한 연구에서는 웹 페이지를 상위 수준에서는 전체 요약을 보여주고, 하위 수준에서 상세 사항을 보여 주도록 다중 수준의 계층 구조로 재조직한다. 특히, [22]에서는 웹 페이지 파싱을 통해서 DOM 트리에 기반한 페이지의 계층적 뷰를 생성한다. 그러나 이러한 연구는 웹 항목 단위의 추출을 고려하지 않는다.

2장에서는 논문에서 참조하는 웹 항목 추출 시스템을 간략하게 소개한다. 3장에서는 인터넷 쇼핑물로부터 관심 있는 상품 항목을 직접 검색하는데 필요한 모든 정보를 표현하는 래퍼를 설계하고, 4장에서는 인터넷 쇼핑물을 이용하는 M-commerce 콘텐츠 제공자 시스템(앞으로 간략하게 M-commerce 시스템이라 부른다)을 설계한다. 5장에서는 시스템의 구현 사항에 대해서 설명하고, 6장에서는 구현된 시스템을 특정 인터넷 쇼핑물 사이트에 대해서 적용하고, 시스템을 데이터 전송량의 관점에서 평가한다. 마지막으로 7장에서는 결론을 언급한다.

2. 웹 항목 추출 시스템

논문에서 인터넷 쇼핑물을 이용하여 M-commerce 시스템 개발시에 [13]의 웹 항목 추출 시스템을 참조한다. 따라서 여기서는 웹 항목 추출 시스템을 간략히 소개한다. 이 시스템은 크게 래퍼 생성기와 웹 항목 추출기의 두 모듈로 구성된다. 래퍼 생성기는 사용자가 GUI(Graphical User Interface)를 사용하여 인터넷상의 임의의 웹 문서로부터 관심 있는 항목 단위의 정보를 추출하는데 필요한 제반 정보

를 생성하여 래퍼로 포장하고, 이를 래퍼 저장소에 저장한다. 사용자가 나중에 이 항목을 참조할 때, 웹 항목 추출기가 사용된다. 웹 항목 추출기는 해당 래퍼를 래퍼 저장소로부터 가져오고, 래퍼에 포함된 항목 추출 정보를 이용하여 해당 항목을 웹으로부터 가져와서 사용자에게 전달한다. 따라서 사용자는 웹 브라우저에 의한 웹 문서 검색 과정 없이 관심 있는 항목을 직접 검색할 수 있게 된다. 시스템은 날씨나 주식 정보와 같이 빈번한 모니터링이 요구되는 웹 정보에 관리에 유용하게 적용될 수 있다.

래퍼 생성기가 구성하는 래퍼는 그림 1의 구조를 갖는다. 래퍼는 그 식별자인 이름(Name)과 EPath들로 구성된다. 각 EPath는 웹 문서상에서 한 개의 웹 항목을 추출하는데 필요한 제반 정보를 표현한다. 이러한 정보에는 항목의 이름(Name), 항목이 포함된 HTML 웹 문서 주소(Target Address), 그리고 문서상의 항목의 경로를 포함한다. 항목 경로는 노드(Node)들의 시퀀스로 표현된다. 여기서 Node는 항목이 포함된 HTML 문서를 트리 구조로 표현할 때, 트리의 루트부터 항목까지의 경로상의 한 노드를 표현한다. Node는 TagName과 TagIndex의 두 속성을 갖는데, TagName에는 HTML 태그 이름을 표현하고, TagIndex에는 해당 태그가 위치한 트리 수준에서 왼쪽부터의 자식 노드 순서를 나타낸다. 즉, 한 개의 Node는 HTML 문서의 트리상의 한 노드를 유일하게 표현한다. 따라서 EPath에 표현된 Node들의 시퀀스를 통해서 문서의 루트 노드로부터 해당 항목까지의 트리상의 경로를 표현할 수 있다.

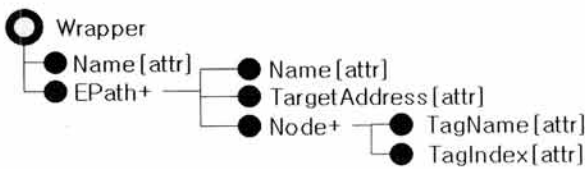


그림 1. 웹 항목 표현 래퍼 구조
Fig. 1. Wrapper structure for web item representation

논문에서는 그림 1의 래퍼 구조에 기반하여 인터넷 쇼핑물의 상품 항목을 효과적으로 표현하는 래퍼 구조를 설계한다. 다음에 래퍼 생성기와 웹 항목 추출기의 알고리즘을 이용하여 인터넷 쇼핑물로부터 상품항목을 래퍼에 등록하고, 나중에 래퍼를 이용하여 항목을 추출하는 M-commerce 시스템을 구성한다.

3. M-commerce 상품 표현 래퍼 구조 설계

여기서는 인터넷 쇼핑물로부터 관심 있는 상품 항목을 검색하는데 필요한 모든 정보를 표현하는 래퍼 구조를 설계한다. 인터넷 쇼핑물로부터 관심 있는 상품 항목을 직접 검색하는데 필요한 정보는 쇼핑물 사이트 주소, 쇼핑물 검색 엔진 주소, 상품 문서 주소 패턴, 관심 있는 상품 항목의 경로 등을 포함한다. 이러한 정보를 표현하는 래퍼 구조를 그림

2에서 보여준다. 이 래퍼 구조에서 루트는 인터넷 쇼핑물을 나타내는 ShopMalls이며 그 자식 노드로 여러 개의 Shop이 존재할 수 있다. 각 Shop 노드는 특정 쇼핑물에 대한 상품 정보를 표현한다. Shop 노드는 해당 쇼핑물의 특정 상품 문서를 검색하는데 필요한 모든 정보를 포함하는 ShopInfo와 그 문서로부터 관심 있는 상품 항목들과 이러한 항목에 대한 경로를 포함하는 ShopItems로 크게 구분된다.

ShopInfo에서, ShopName은 쇼핑물 이름을 나타내고, ShopAddress는 쇼핑물의 인터넷 주소를 나타내고, SearchAddrPattern은 쇼핑물의 검색 엔진에 대한 인터넷 주소 패턴을 나타내고, ItemAddrPattern은 쇼핑물에서 관심 있는 상품 문서에 대한 인터넷 주소 패턴을 나타낸다. 여기서 ShopAddress, SearchAddrPattern, ItemAddrPattern의 정보를 이용하면 쇼핑물의 명세된 상품 분류의 상품 문서에 직접 접근할 수 있다.

ShopItems는 ShopInfo로부터 검색된 상품 문서에 대해서 관심 있는 항목들과 이러한 항목들에 대한 문서상의 경로를 표현한다. 여기서는 문제를 단순화하기 위해서, ShopItems에 관심 있는 상품 항목들을 단지 3가지로 제한한다. 상품 이름을 나타내는 ProductTitle, 상품 제조사를 표현하는 CompanyAlias, CompanyValue, 상품의 가격을 표현하는 CostAlias, CostValue 등이 그것이다. 그러나 사용자는 그림 2의 래퍼 구조를 3MP[14]의 상품 표현모델로 확장할 수 있다.

ShopItems에는 특정 상품 항목에 대해서 2가지 요소가 포함된다. 가령, 상품 제조사를 표현하는데 CompanyAlias와 CompanyValue의 두 요소가 사용된다. CompanyAlias는 상품을 만든 '제조사'의 어휘를 의미하는데, 이는 상품 유형에 따라서 그 어휘가 달리 사용될 수 있다는 사실을 반영한다. 가령, 책의 경우에는 CompanyAlias는 '출판사'가 될 수 있다. CompanyValue는 그 상품을 만든 실제 제조사를 표현한다. 시스템에서 그림 2의 래퍼 구조는 XML 문서로 표현되는데, 이때 Alias 항목은 해당 Value 항목의 속성으로 표현된다.

각 상품 항목은 그림 1의 EPath와 같이 부 요소로 Node들의 시퀀스를 갖는다. 이러한 요소는 연관된 항목에 대한 상품 문서상의 경로를 표현하는데 사용된다. 여기서 경로는 상품 문서를 DOM 트리 구조로 표현하였을 때, 항목은 트리의 한 노드로 표현되는데, 이 노드에 대한 트리 루트로부터의 경로를 의미한다.



그림 2. 인터넷 쇼핑물의 상품항목 표현 래퍼 구조
Fig. 2. Wrapper structure for product item representation

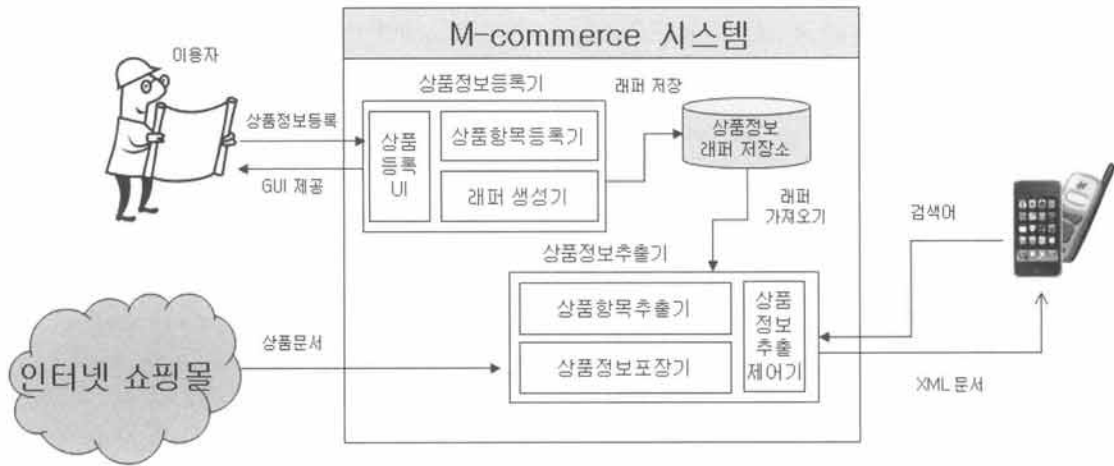


그림 3. M-commerce 시스템 구조
Fig. 3. Architecture for M-commerce system

4. M-commerce 시스템 설계

다음에는 쇼핑물의 관심 있는 상품 항목을 3장에서 설계한 래퍼에 등록하고, 나중에 래퍼에 등록된 상품 항목을 검색하는 기능을 제공하는 M-commerce 시스템을 설계한다. 논문에서 고려하는 M-commerce 시스템의 구조는 그림 3과 같다.

시스템은 쇼핑물의 상품정보등록기, 상품정보추출기, 그리고 상품정보 래퍼 저장소로 구성된다. 이용자는 상품정보등록기의 상품등록 UI를 통해서 사용자가 원하는 인터넷 쇼핑물의 상품정보를 등록하고, 등록된 정보를 그림 2의 래퍼 구조를 갖는 XML 문서[23]로 생성하여 상품정보 래퍼 저장소에 저장한다. 사용자가 휴대폰이나 스마트폰 등의 모바일 기기를 이용하여 상품정보를 요청하면, 상품정보추출 제어기(controller)는 상품정보 래퍼 저장소로부터 해당 상품정보 래퍼를 가져와서 상품항목추출기에 전달한다. 상품항목추출기는 래퍼의 ShopInfo를 이용하여 해당 상품 문서들을 인터넷으로부터 가져오고, ShopItems를 이용하여 이러한 문서 상에서 상품 항목들을 추출하여 상품정보포장기에 전달한다. 상품정보포장기는 추출된 상품 항목들을 포함하는 XML 문서를 생성하고, 상품정보추출 제어기는 생성된 XML 문서를 사용자에게 전달한다.

상품정보등록기는 상품항목등록기, 래퍼 생성기, 상품등록 UI의 3 부분으로 구성된다. 상품등록 UI는 이용자가 관심 있는 인터넷 쇼핑물로부터 관심 있는 상품 항목을 등록하는데 편리한 GUI(Graphical User Interface)를 제공한다. 이용자는 먼저, 상품등록 UI 상에서 웹 브라우저를 띄워서 이 웹 브라우저 상에서 관심 있는 쇼핑물과 관심 있는 상품 문서에 차례대로 접근하고, 상품 문서에 포함된 관심 있는 항목을 표시한다. 다음에, 상품항목등록기는 래퍼 생성기를 이용하여 문서상에 표시된 항목에 대한 경로 시퀀스를 생성하고, 이를 그림 2의 구조를 갖는 래퍼로 포장하여 상품정보 래퍼 저장소에 저장한다.

그림 3에서 상품정보등록기의 래퍼 생성기와 상품정보추출기의 상품항목추출기는 [13]의 래퍼 생성기와 웹 항목 추출기의 컴포넌트를 각각 이용하여 구현된다. 따라서 여기서는 상품정보등록기 중심으로 시스템을 고려한다.

그림 4는 상품정보등록기에 대해서 설계한 UML 클래스 다이어그램을 보여준다. ManagerUI, ManagerClientControl, WrapperCreator는 각각 상품등록 UI, 상품항목등록기, 래퍼 생성기를 나타낸다. ManagerUI는 이용자가 웹 브라우저를 통해서 특정 쇼핑물 및 상품 문서에 접근하여 상품 문서에 포함된 관심 있는 항목을 추출하는데 편리한 메뉴를 제공한다. 사용자는 이러한 메뉴를 통해서 상품 문서 상에서 관심 있는 항목을 마우스로 드래그하여 블록으로 선택한다.

ManagerClientControl은 ManagerUI를 통해서 이용자가 선택한 항목들을 래퍼에 등록한다. 이용자는 ManagerUI를 통해서 항목을 선택하고, PathExtractor는 선택된 각 항목에 대한 상품 문서상의 경로인 PEPATH를 WrapperCreator를 이용하여 생성한다. WrapperCreator는 [13]에서 제공된 래퍼 생성기 컴포넌트이며, 선택된 항목에 대한 문서 상의 경로를 생성한다. PePathList는 선택된 모든 항목들에 대한 PEPATH들을 포함한다. ManagerClientControl은 SaveShop Wrapper 메소드를 이용하여 PePathList에 포함된 항목들을 포함하는 그림 2의 래퍼를 생성하고, 이를 상품정보 래퍼 저장소에 저장한다. ShopInfo, PePathList, PEPATH, Node는 XmlElementLoader를 상속받는데, 이는 래퍼를 XML 문서로 생성하기 위함이다. 또한, ManagerClientControl은 ManagerUI를 통해서 사용자가 검색한 쇼핑물 및 상품 문서에 대한 정보를 ShopInfo 객체에 저장하는데, 이 객체는 래퍼 생성시 그림 2의 ShopInfo 요소를 구성하는데 사용된다.

그림 5는 상품정보등록기에 대한 UML 시퀀스 다이어그램을 보여준다. ManagerUI는 ManagerClientControl을 생성하고, ManagerClientControl은 ShopInfo를 생성한 후에, PePathList를 생성한다. PePathList는 PathExtractor를 생성하여 선택된 항목에 대한 문서상의 경로를 생성할 준비를

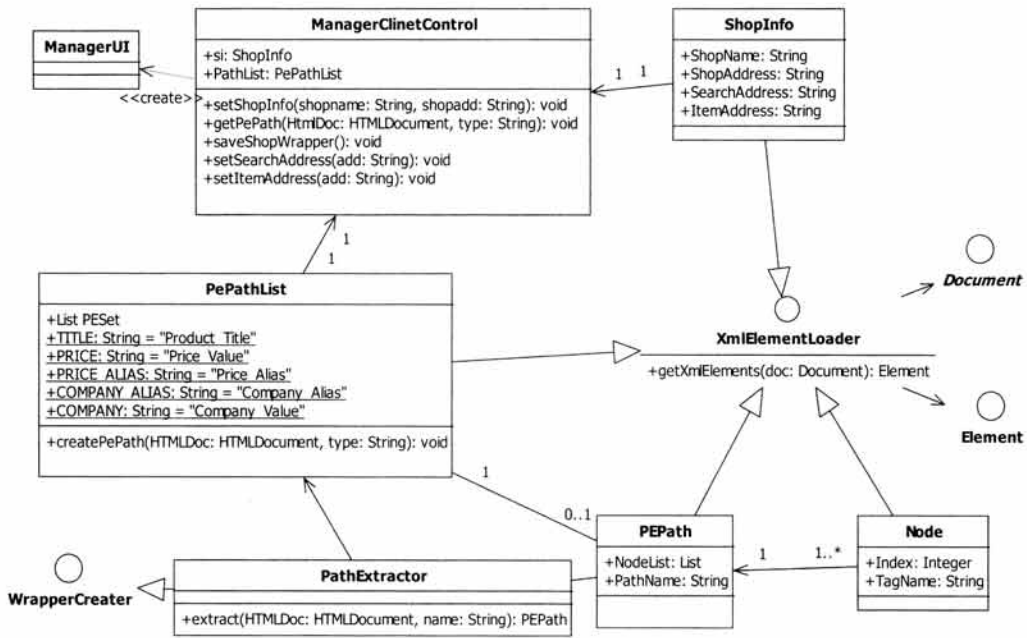


그림 4. 상품정보등록기의 UML 클래스 다이어그램
Fig. 4. UML class diagram of product information register

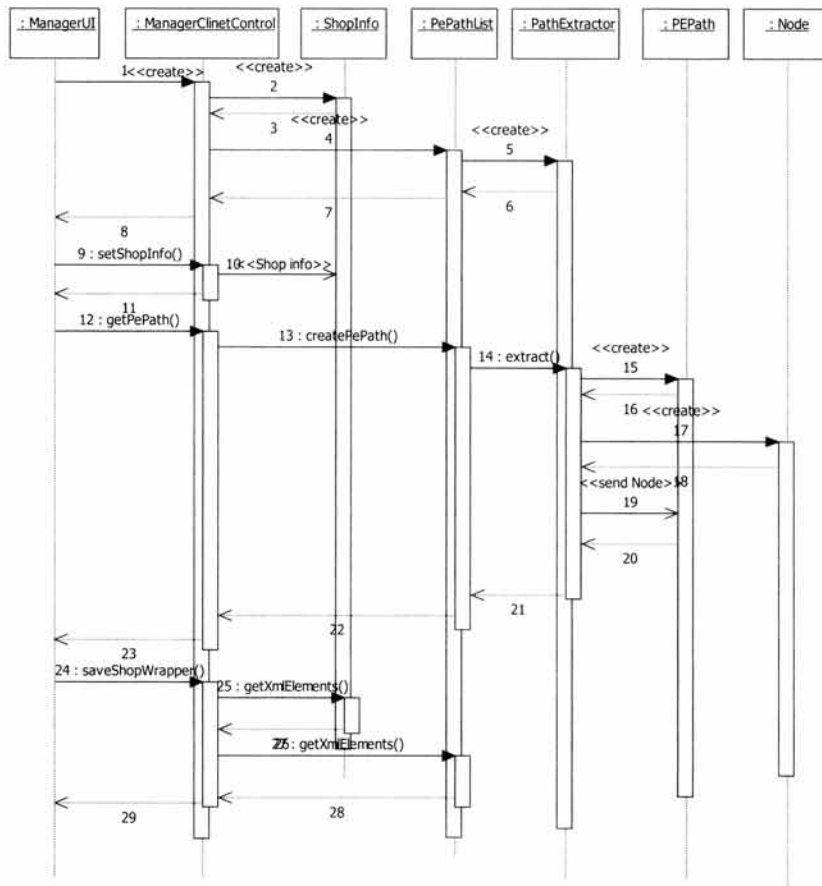


그림 5. 상품정보등록기의 UML 시퀀스 다이어그램
Fig. 5. UML sequence diagram of product information register

한다. 다음에, 쇼핑몰 사용자로부터 쇼핑몰의 정보가 SetShopInfo를 통하여 ShopInfo에 저장된다.

쇼핑몰 사용자는 ManagerUI를 통해서 상품 항목을 선택하고, ManagerClientControl의 getPePath를 실행한다. getPePath는 PePathList의 createPePath를 실행하고, PePathList는 사용자가 선택한 항목에 대해서 문서상의 경로인 PEPATH를 WrapperCreator를 이용하여 생성하고, 이들을 리스트로 구성한다. 사용자가 관심 있는 모든 항목을 선택하였으면, SaveShopWrapper는 구성된 PePathList를 래퍼로 생성한다. 이때, ShopInfo와 PePathList의 getXmlElement는 래퍼에 포함되는 해당 XML 요소를 생성한다.

5. 구현

4장에서 설계된 M-commerce 시스템은 기본적으로 Java 플랫폼 상에서 컴포넌트 기반으로 구현되었다. 인터넷 쇼핑몰의 상품 문서가 대부분 HTML 문서로 작성되어 있기 때문에 이러한 문서에 접근하여 관심 있는 항목을 용이하게 추출하기 위해서 MS Internet SDK[24]를 이용하였다. MS Internet SDK를 사용하는 모듈들은 .NET 플랫폼 상에서 개발되었으며, Java 플랫폼 상에서 개발된 다른 모듈과의 연동을 위해서 웹서비스로 구성하였다. 이와 함께, .NET 프레임워크의 웹 브라우저인 System.web의 WebBrowser[25]를 사용하여 쇼핑몰 이용자에 의한 HTML 문서 작업을 지원하였다.

6. 적용 및 평가

여기서는 앞서 구현된 M-commerce 시스템을 실제의 인터넷 쇼핑몰에 대해서 적용한다. 우리는 이 시스템을 사용하여 인터넷 쇼핑몰인 인터파크[26]로부터 관심 있는 상품 정보를 추출하여 래퍼로 생성하는 과정을 단계적으로 보여

준다. 그림 6은 쇼핑몰 이용자가 시스템의 상품등록 UI를 통해서 인터파크 쇼핑몰에 접속하는 과정을 보여준다. 이를 위해서, UI의 오른쪽 상단에 위치한 두 입력란에 각각 쇼핑몰의 이름과 주소를 입력한다. 입력된 쇼핑몰의 이름과 주소는 나중에 생성될 래퍼의 ShopName과 ShopAddress에 각각 저장된다.

다음에, 쇼핑몰에서 제공하는 검색 엔진을 사용하여 특정 상품 문서를 검색한다. 그림 7은 쇼핑몰의 검색 엔진을 이용하여 상품을 검색 과정을 보여준다. 상품등록 UI의 하단에 위치한 '검색 주소'란의 우측 끝에 위치한 '현재 페이지'를 클릭하면 '검색 주소'란에 쇼핑몰 검색 엔진 주소가 나타난다. 이러한 주소에서 쇼핑몰의 검색 엔진 주소임을 알려주는 부분만 남기고 나머지는 삭제한다. 그 결과의 주소는 쇼핑몰 검색 엔진의 주소 패턴이 된다. 다음에 상품등록 UI의 우측 끝의 아래에 위치한 '입력'을 누르면 이 주소 패턴이 그림 2의 래퍼의 SearchAddrPattern에 저장된다.

다음에, 쇼핑몰의 검색 엔진을 이용하여 관심 있는 특정 상품을 선택한다. 상품등록 UI의 하단에 위치한 '물품 주소'란의 우측 끝에 위치한 '현재 페이지'를 클릭하면 '물품 주소'란에 해당 상품 문서의 주소가 나타난다. 이러한 주소에서 상품 문서의 주소임을 알려주는 부분만 남기고 나머지는 삭제한다. 그 결과의 주소는 상품 문서의 주소 패턴이 된다. 다음에 상품등록 UI의 우측 끝의 아래에 위치한 '입력'을 누르면 이 주소 패턴이 래퍼의 ItemAddrPattern에 저장된다. 그림 8은 상품 문서에 대한 주소 패턴을 입력하는 화면을 보여준다.

다음에, 사용자는 검색된 상품 문서에서 관심 있는 항목을 추출하여 래퍼에 등록한다. 그림 9는 이러한 과정을 보여준다. 쇼핑몰 이용자는 관심 있는 항목을 마우스로 드래그하여 블록으로 표시하고, 상품등록 UI의 상단 메뉴에서 '추출' 항목을 클릭하면 나타난 항목 리스트 가운데서 등록하고자 하는 항목을 풀다운메뉴로부터 선택한다. 그림 9의 경우에, 사용자가 상품 이름을 추출하고자 하므로 '추출' 항



그림 6. 시스템 UI를 사용한 쇼핑몰 접속 화면
Fig. 6. Access to shopping malls using product registration UI

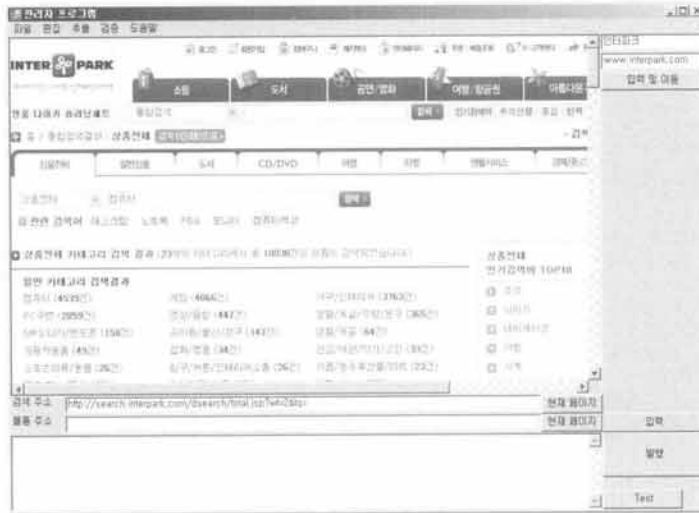


그림 7. 쇼핑몰 검색엔진 사용을 위한 검색 주소 패턴 입력
Fig. 7. Input of the address pattern of search engine



그림 8. 쇼핑몰의 상품 문서 물품 주소 패턴 입력
Fig. 8. Input of the address pattern of product documents



그림 9. 상품 항목 추출 과정 화면
Fig. 9. Extraction of product items

```

<?xml version="1.0" encoding="utf-8"?>
<ShopMalls>
  <Shop>
    <ShopInfo>
      <ShopName>인터파크</Name>
      <ShopAddress>www.interpark.com</ShopAddress>
      <SearchAddrPattern>http://search.interpark.com/dsearch/total.jsp?wf=2&amp;:tq=</SearchAddrPattern>
      <ItemAddrPattern>http://www.interpark.com/product/MallDisplay.do?</ItemAddrPattern>
    <ShopItems>
      <ProductTitle>
        <Node Name="CENTER" Index="19" />
        <Node Name="DIV" Index="0" />
        <Node Name="TABLE" Index="18" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
        <Node Name="TABLE" Index="1" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="1" />
        <Node Name="TD" Index="0" />
      </ProductTitle>
      <CompanyAlias>
        <Node Name="CENTER" Index="19" />
        <Node Name="DIV" Index="0" />
        <Node Name="TABLE" Index="18" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
        <Node Name="TABLE" Index="7" />
        <Node Name="TBODY" Index="1" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="0" />
      </CompanyAlias>
      <CompanyValue>
        <Node Name="CENTER" Index="19" />
        <Node Name="DIV" Index="0" />
        <Node Name="TABLE" Index="18" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
        <Node Name="TABLE" Index="7" />
        <Node Name="TBODY" Index="1" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
      </CompanyValue>
      <CostAlias>
        <Node Name="CENTER" Index="19" />
        <Node Name="DIV" Index="0" />
        <Node Name="TABLE" Index="18" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
        <Node Name="TABLE" Index="5" />
        <Node Name="TBODY" Index="1" />
        <Node Name="TR" Index="2" />
        <Node Name="TD" Index="0" />
      </CostAlias>
      <CostValue>
        <Node Name="CENTER" Index="19" />
        <Node Name="DIV" Index="0" />
        <Node Name="TABLE" Index="18" />
        <Node Name="TBODY" Index="0" />
        <Node Name="TR" Index="0" />
        <Node Name="TD" Index="1" />
        <Node Name="TABLE" Index="5" />
        <Node Name="TBODY" Index="1" />
        <Node Name="TR" Index="2" />
        <Node Name="TD" Index="1" />
        <Node Name="SPAN" Index="1" />
      </CostValue>
    </ShopItems>
  </Shop>
  <Shop>
    ...
  </Shop>
</ShopMalls>

```

그림 10. 상품 문서로부터 생성된 래퍼
 Fig. 10. Wrapper generated from product documents

목의 리스트에서 '제품명'을 선택한다. 마찬가지로 방법으로, 상품에 대한 제조사의 별칭과 값, 가격의 별칭과 값에 대한 항목을 상품 문서로부터 추출한다. 이와 같이 상품 문서에 대해서 관심 있는 항목이 모두 추출되었으면, 상품등록 UI의 오른쪽 하단에 위치한 '발행' 버튼을 누른다. 그러면 상품 문서로부터 추출된 항목들을 포함한 래퍼가 생성되어 상품 정보 래퍼저장소에 저장된다.

그림 10은 위의 과정을 통해서 생성된 래퍼를 보여준다. 여기서 <productTitle> 상에 표현된 경로 시퀀스는 그림 9에서 마우스로 선택된 항목에 대한 문서의 BODY 태그로부터의 해당 항목에 이르는 경로를 나타낸다. CompanyAlias 등의 다른 태그 상에 표현된 경로 시퀀스도 마찬가지로 해당 항목에 대한 문서상의 경로를 나타낸다.

다음에는, 위와 같이 인터넷 쇼핑물에 적용한 M-commerce 시스템의 성능을 평가한다. 평가 방법으로, 인터넷 쇼핑물상의 상품으로부터 관심 있는 항목을 검색하는 과정에서 사용자의 모바일 폰에 전송되는 데이터 양의 관점에서 일반 웹 브라우저를 사용했을 때와 논문의 시스템을 사용했을 때의 두 가지 경우를 비교한다. 평가를 위해서 적용된 인터넷 쇼핑물은 인터파크이고, 상품 검색어는 "컴퓨터"이고, 관심 있는 상품 항목은 상품명, 제조사, 가격이다.

표 1. 웹 브라우저와 시스템 비교
Table 1. Comparison between web browser and system

구분		웹 브라우저	시스템
검색 데이터	HTML 코드	116,880	×
	확장 코드	174,034	×
	이미지	3,993,597	×
	소계	4,284,511	×
상품 데이터	HTML 코드	128,542	14,519
	확장 코드	369,207	×
	이미지	4,188,715	×
	소계	4,686,464	14,519
계		8,970,975	14,519

표 1은 웹 브라우저 대비 시스템의 평가 결과를 보여준다. 상품 검색시 전달되는 데이터는 '검색 데이터'와 '상품 데이터'로 구분한다. '검색 데이터'는 쇼핑물의 검색 엔진을 사용하여 해당 상품 문서를 검색하는 과정에서 전송되는 데이터이며, '상품 데이터'는 해당 상품 문서에 포함된 데이터를 의미한다. 또한, 검색 데이터와 상품 데이터에 포함된 데이터는 HTML 코드, 확장 코드, 이미지의 3가지로 구분된다. HTML 코드는 문서에 포함된 HTML 소스 코드로 구성된 데이터이며, 확장 코드는 문서에 포함된 CSS, JavaScript 코드 등으로 구성된 데이터이며, 이미지는 문서에 포함된 이미지의 바이트 단위 용량을 의미한다.

일반 웹 브라우저를 통해서 상품을 검색할 경우에는 검색 데이터와 상품 데이터에 포함된 모든 데이터가 사용자 기기로 전송되는 반면에, 시스템의 경우에는 검색 데이터는 전

혀 전송되지 않으며, 단지 상품 데이터만 고려된다. 이 경우에도 확장 코드와 이미지도 전송되지 않고, 단지 HTML 코드만이 고려될 수 있다. 또한 HTML 코드 전체가 전송되지 않고, 단지 관심 있는 항목의 데이터만이 사용자 기기로 전송된다. 그 결과로, 일반 웹 브라우저의 경우에 검색 데이터가 약 4.0MB, 상품 데이터가 약 4.5MB이며, 합계 8.5 MB가 사용자 기기로 전송되는 반면에, 시스템을 적용한 경우에는 단지 관심 있는 상품 항목만이 전송되어 14.2KB만이 사용자 기기로 전송된다. 결과적으로, 일반 브라우저를 사용하면 논문의 시스템을 사용하는 경우에 비해서 관심 있는 상품 정보를 획득하는데 필요한 데이터가 약 605.5배 더 많이 전송된다.

7. 결 론

논문에서는 인터넷 쇼핑물을 이용하는 M-commerce 시스템을 설계하고 구현하였다. 이 시스템을 인터넷 쇼핑물에 적용한 결과는 일반 웹 브라우저를 사용한 경우와 비교해서 관심 있는 상품정보를 획득하는데 필요한 데이터 전송량이 대폭적으로 감소되었음을 보여준다. 그 결과로, 불필요한 데이터 전송에 따른 통신 비용이 상당히 절감될 수 있다. 또한, 모바일 폰 상에서 사용자의 상품 검색 편의성이 크게 향상된다. 이는 래퍼를 통해서 일련의 검색과정 없이 직접 상품 문서에 접근하여 사용자에게 관심 있는 상품 항목만을 추출하기 때문이다. 이러한 데이터 전송량 감소 및 통신 비용 절감 효과는 개인화(personalization)를 통해서 더욱 향상시킬 수 있다. 사용자의 프로파일에 기반하여 상품 문서들을 먼저 필터링하면, 사용자는 '관심 있는 상품'에 대한 관심 있는 항목만을 검색하면 된다.

최근에 삼성 갤럭시 S3와 같이 쿼드코어까지 탑재한 성능이 높은 스마트 폰이 속속 출시되고 있다. 따라서 향후 개선 및 연구과제는 본 논문의 시스템에서 상품정보등록기의 상품등록 UI 모듈을 스마트 폰에 탑재하면, 서버상의 상품정보등록기와 연동하는 방식으로 이용자가 스마트폰 상에서 M-commerce 쇼핑물을 손쉽게 구축하는 것도 가능해질 것이다.

참 고 문 헌

- [1] H. Wilcox, "Mobile Payment Strategies - Opportunities & Markets 2011-2015," Juniper Research, 2011.
- [2] 스마트폰 2천만시대의 모바일 쇼핑현황 [Internet], [http://mobizen.pe.kr/1130\(2011.11.16.\)](http://mobizen.pe.kr/1130(2011.11.16.))
- [3] N. Sadeh, "M-commerce: Technologies, Services, and Bussiness Models, Reading," 2002, Wiley.
- [4] J. J. Wang, Z. Song, P. Lei, R. E. Sheriff, "Design and Evaluation of M-commerce Applications," Asia-Pacific Conference on Communications, Oct. 2005.
- [5] Webraska [Internet], <http://www.webraska.com/>

- [6] Vodafone [Internet], <http://www.vodafone.com/>, Orange, Telefonica Moviles, T-Mobile, and Vodafone form a new association to drive interoperable mobile payments, 2003.
- [7] MONETA [Internet], <http://www.sktelecom.com/>
- [8] NeMo [Internet], <http://www.kt.com/>
- [9] Google wallet [Internet], <http://www.google.com/wallet/>
- [10] F-Commerce [Internet], <http://www.facebook.com/likecommerce>
- [11] Smart Wallet [Internet], <http://www.smartwallet.co.kr/>
- [12] 모바일 11번가 [Internet], <http://m.11st.co.kr/>
- [13] J. Hur and S. Ha, "GUI Based Generalized Fine-Grain Web Information Extraction", Proceedings of Korea Computer Congress, 2006. 6. 21.
- [14] S. Ha, "Design and Implementation of a Mobile System for Exploiting the Internet Product Information Effectively", The Korea Information Processing Society Transactions Part D, Vol.12, No.3, p.493-498, 2005.
- [15] S. Ha and K. Kim, "Design and Application of a XML Based Product Catalog", The Korea Information Processing Society Transactions Part D, Vol.9, No.3, pp.523-530, 2002.
- [16] XSL Transformations (XSLT) [Internet], <http://www.w3.org/TR/xslt>
- [17] Z. Liu, W. K. Ng, E. P. Lim, "Personalized Web Views for Multilingual Web Sources," IEEE Internet Computing, pp.16-22, July. August 2004.
- [18] J. Ferie, B. Kumer, D. Lieuwen, "WebViews: Accessing Personalized Web Content and Services," ACM 1-58113-348-0, pp.576-586, May 1-5, 2001.
- [19] J. Yang, J. Choi, "Knowledge-Based Wrapper Induction for Intelligent Web Information Extraction," Springer, Web Intelligence, No.8, pp.153-172, 2003.
- [20] E. Lank and S. Phan, "Focus+context sketching on a pocket PC," in Proc. of ACM CHI, pp.1275-1278, 2004.
- [21] C. C. Yang and F. L. Wang, "Fractal summarization for mobile devices to access large documents on the Web," In Proc. o the international WWW conference, pp.215-224, 2003.
- [22] D. Zhang, "Web Content Adaptation for mobile handheld devices," Communications of the ACM, pp.75-79, 2007.
- [23] XML[Internet], <http://www.w3.org/TR/xml/>
- [24] Microsoft Internet SDK, "About MSHTML" [Internet], <http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/mshtml/overview/overview.asp>
- [25] System.web, MSDN [Internet], <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemweb.asp>
- [26] INTERPARK [Internet], <http://www.interpark.com>



하 상 호

e-mail : hsh@sch.ac.kr

1988년 서울대학교 계산통계학과(학사)

1991년 서울대학교 계산통계학과
(이학석사)

1995년 서울대학교 전산학과(이학박사)

1995년~1996년 한국전자통신연구원

PostDoc.

1996년~1997년 미국 MIT PostDoc.

2005년~2006년 미국아이오와주립대학 방문 교수

1997년~현 재 순천향대학교 컴퓨터공학과 교수

관심분야: 프로그래밍언어, 유비쿼터스 컴퓨팅, M-commerce, u-Health 등