

모바일 클라이언트를 위한 상품정보 객체계층구조 구성

(Construction of Product Information Hierarchy for Mobile Clients)

하 상 호^{*}
(Sangho Ha)

요약 무선 인터넷 기술과 모바일 컴퓨팅 기술의 발달로, M-Commerce가 PDA나 모바일 폰 등과 같은 다양한 무선 장치상에서 실현되고 있다. 현재 M-Commerce를 위한 상품 컨텐츠는 목표 무선 장치의 특징을 충족할 수 있도록 새롭게 작성되고 있다. 이것은 자원의 중복이며, 자원 관리에 엄청난 노력이 요구된다. 따라서 현재 E-Commerce를 위해서 제공되고 있는 인터넷 상품 정보를 효과적으로 이용하는 것이 필요하다. 그러나 이러한 이용에는 적은 메모리 용량, 작은 디스플레이 화면, 낮은 수행속도 등의 모바일 장치의 많은 제약성이 고려되어야 한다. 본 논문에서는 모바일 폰 상에 인터넷 XML 상품 문서를 효과적으로 사용하기 위해 클라이언트에서 문서에 대한 객체계층구조를 구성하는 4가지 방법을 제안하고, 실험을 통해서 사용자 응답시간과 메모리 사용량 관점에서 비교, 분석한다.

키워드 : M-Commerce, 상품표현 모델, 객체계층구조, 모바일 폰, XML, Java

Abstract With the advances of wireless technologies and mobile computing, m-commerce is being realized on many kinds of mobile devices. Service contents for m-commerce are usually newly written to meet specific characteristics of user's mobile devices, thus requiring formidable efforts. So, it is very important to effectively exploit the Internet product information currently being provided for e-commerce. However, bringing those Internet contents to mobile devices is far from straightforward due to the limitations of mobile devices such as little memory, small displays, low processing speeds, and so forth. In this paper, assuming that the Internet products are represented in XML, we suggest four methods to construct the object hierarchy for effectively viewing the documents on the mobile devices. We then compare and analyze them by experiments in terms of response times and required memory size.

Key words : M-Commerce, Product Model, Hierarchy of Objects, Mobile Phones, XML, Java

1. 서론

오늘날 인터넷의 광범위한 보급은 이제 유선을 넘어 무선 이동통신 기술로의 급발전을 가져오게 되었고, 생활에 있어서도 무선단말기를 통한 인터넷상에서의 정보검색이 시간과 장소를 구애받지 않고 가능하게 되었다. 2001년 말, 전체인구의 14%에 달하는 8억 5500만의 인구가 모바일 폰을 사용하고 있으며, 이러한 폭발적인 증가에 힘입어 M-Commerce가 차세대 상거래의 형태가 될 것으로 예고된다[1].

이러한 추세에 발맞추어 NTT DoCoMo의 i-Mode 포탈[2], Nordea의 WAP 모바일 banking 서비스[3], Web-

raska의 스마트존 플랫폼[4], SK텔레콤의 네이트[5], KTF의 매직엔[6], LG텔레콤의 이지아이[7] 등과 같이 국내·외의 많은 업체들이 M-Commerce를 위한 서비스를 제공하고 있지만, E-Commerce에서와 같이 풍부한 상품정보가 서비스되지 않는 실정이다. 현재, 무선 단말기에 제공되고 있는 상품정보 서비스는 업체별로 별도의 무선 사이트를 구축하여 제공되고 있어 무선 단말기간에 호환되지 않으며, E-commerce의 상품정보와도 호환되지 않고 있다. 향후 E-Commerce와 M-Commerce가 공존하게 될 경우, 중복을 통한 자원의 낭비 등 많은 문제점을 안고 있다.

무선 사이트 구축을 통하지 않고 웹 문서를 모바일 단말기의 작은 화면에 디스플레이하기 위한 많은 연구[8-13]가 이루어져 왔다. AvantGo[8]와 WebViewer[9]은 웹 문서를 단지 단말기 화면에 디스플레이하기에 적

* 중신회원 : 순천향대학교 컴퓨터학부 교수

hsh@sch.ac.kr

논문접수 : 2004년 3월 4일

심사완료 : 2004년 11월 17일

합한 형식으로 변환한다. 이러한 변환은 프락시 서버(proxy server)에서 이루어지며, 변환 결과는 화면 크기와 네트워크 프로토콜의 제한으로 인해 화면들의 집합으로 구성되어 화면간의 빈번한 이동이 필요하다. [10, 11]에서는 주로 웹 문서에 포함된 링크 항목들을 추출하여 그 문서의 개략적인 전체 구조를 먼저 보여주고, 사용자가 특정 링크를 선택하면, 링크에 연결된 세부 내용을 적절히 디스플레이한다. [12]에서는 웹 문서에서 추출된 관심 있는 항목들을 작은 화면에 효과적으로 디스플레이할 수 있는 여러 방법들을 제시하고 평가한다. IBM의 Transcoding[13]은 웹 문서를 목표 단말기에 효과적으로 디스플레이하기 위해서 문서를 적용시키고, 형식을 변환시키고, 내용을 필터링하고, 재구성한다. 위의 모든 연구들은 HTML 웹 문서를 모바일 폰의 작은 화면에 효과적으로 디스플레이하기 위한 일반적인 적용에 관하며, 구조적 문서인 상품정보에 관해서는 고려하지 않는다. 상품정보는 일반적인 HTML 문서와는 달리 구조적 문서이며, 이러한 구조적 문서를 작은 화면에 효과적으로, 그리고 친숙하게 사용자에게 상품정보를 제공하는 것이 필요하다.

[14]의 연구에서, 인터넷상의 상품정보를 모바일 폰에서 효과적으로 디스플레이 할 수 있는 시스템을 개발하고 있다. 이 시스템은 서버/클라이언트 구조로서, 클라이언트가 상품 검색을 요청하면, 이 요청이 서버에 전달되고, 서버는 클라이언트의 요청을 해석하여, 인터넷 검색을 통하여 해당 상품 문서를 가져오고, 이 문서를 3MP(Merchant Model for Mobile Phones) 모델[8]의 구조를 갖는 XML 문서로 변환하여 클라이언트에 전달한다. 논문에서는 인터넷 상품정보가 표준 문서 기술 언어인 XML로 기술되어 있다고 가정한다. 그러나 이러한 가정은 일반성을 상실하지 않는다. 왜냐하면, HTML 문서는 XML 문서로 변환될 수 있기 때문이다[15]. 3MP 모델은 클라이언트의 모바일 폰상에 상품정보를 효과적으로 표현하도록 설계된 XML 기반 상품표현모델이다. 클라이언트는 전달된 문서를 파싱하여 DOM 트리를 생성하고, 이 트리로부터 3MP 모델을 반영하는 상품정보 객체계층구조(Hierarchy of objects)를 구성하고, 이 구조를 이용하여 사용자의 반응에 따라서 4단계에 걸쳐서 상품정보를 모바일 폰의 화면상에 효과적으로 디스플레이한다.

이 시스템이 Java와 XML 기술을 사용하여 구현되어 서버나 클라이언트의 특정 기종에 독립되어 사용될 수 있다는 장점을 갖는다. 그러나 클라이언트에서 XML 상품 문서에 대해서 구성되는 객체계층구조는 문서의 크기가 큰 경우에 사용자의 모바일 폰이 효과적으로 다루기에는 그 메모리 용량이나 처리 능력이 미흡하다. 본문에서는 객체계층구조를 구성하는 방법 차원에서 이

러한 문제를 해결할 수 있는 한 가지 방안을 제시한다. 먼저, 클라이언트가 XML DOM 트리로부터 객체계층구조를 생성하는 여러 가지 방법을 제안하고, 다음에 사용자 응답시간과 메모리 사용량의 관점에서 제안된 객체계층구조 구성 방법을 비교, 분석한다.

논문의 구성은 다음과 같다. 2장에서는 논문의 배경으로, 인터넷 상의 상품정보를 효과적으로 이용하기 위해 설계된 3MP 모델과 클라이언트/서버 시스템 구조, 클라이언트가 서버로부터 전달된 XML 상품정보를 디스플레이하는 방법 등에 대해서 간략하게 살펴보고, 3장에서는 클라이언트에서 XML DOM 트리로부터 객체계층구조를 구성하는 4가지 방법을 제안한다. 4장에서는 실험을 통해서 제안된 4가지의 객체계층구조 구성 방법을 사용자 응답시간과 메모리 사용량 관점에서 비교, 분석한다. 마지막으로 5장에서는 결론을 언급한다.

2. 배경

여기서는 논문의 배경으로, [14]에서 연구된 3MP 상품 표현모델, 모바일 클라이언트에 웹 상품정보를 제공하는 시스템 구조, 그리고 모바일 클라이언트에서 상품정보를 디스플레이하는 방법에 대해서 간략하게 설명한다. 3MP(Merchant Model for Mobile Phones) 상품표현 모델은 인터넷 상의 상품정보를 모바일 폰상에 효과적으로 디스플레이될 수 있도록 설계된 상품표현 모델이다. 이 모델은 E-Commerce를 위해서 설계된 전자 카탈로그[16]의 슬림 버전이며, 대부분의 고객이 관심을 갖는 필수적인 항목만을 뽑아내어 정리한 것이다. 그림 1은 이 3MP 모델을 XML DTD로 표현한 것이다. 상

```
<?xml version="1.0" encoding="UTF-8"?>
<ELEMENT ProductCatalog (Product)*>
<ELEMENT Product (ClassIdent, ProductData)>
<ELEMENT ClassIdent (Class, Ident, CountryofOrigin)>
<ELEMENT Class (Value, Class?)*>
<ELEMENT Value (#PCDATA)>
<ELEMENT Ident (#PCDATA)>
<ELEMENT CountryofOrigin (#PCDATA)>
<ELEMENT ProductData (Name, Image, Description, Price,
Specification, Manufacture, Manufacturer,
Participant, Evaluation)*>
<ELEMENT Name (#PCDATA)>
<ELEMENT Image (Description?)*>
<ATTLIST Image
type CDATA #REQUIRED
source CDATA #REQUIRED>
<ELEMENT Description (#PCDATA)>
<ELEMENT Price (Amount, DiscountRate, Company)*>
<ATTLIST Price
type CDATA #REQUIRED>
<ELEMENT Amount (#PCDATA)>
<ATTLIST Amount
currency CDATA #REQUIRED>
<ELEMENT DiscountRate (#PCDATA)>
<ELEMENT Company (#PCDATA)>
<ELEMENT Specification (#PCDATA)>
<ATTLIST Specification
type CDATA #REQUIRED
name CDATA #REQUIRED>
<ELEMENT Manufacture (#PCDATA)>
<ELEMENT Manufacturer (#PCDATA)>
<ELEMENT Participant (#PCDATA)>
<ATTLIST Participant
role CDATA #REQUIRED>
<ELEMENT Evaluation (Evaluator, Description)*>
<ELEMENT Evaluator (#PCDATA)>
```

그림 1 XML DTD로 표현된 3MP 모델

품 정보는 분류·식별(ClassIdent) 정보와 상품의 본질적 정보(ProductData)로 구분되고, ClassIdent에는 상품을 분류하고, 식별할 수 있는 정보가 표현되고, ProductData에는 상품 이름(Name), 이미지(Image), 상품 개요(Description), 다양한 가격정보(Price), 상품 명세(Specification), 제조사(Manufacturer), 제조일(Manufacture), 제작자(Participant), 평가정보(Evaluation) 등이 표현된다.

3MP 모델은 모바일 폰을 사용한 M-Commerce를 위한 상품정보 콘텐츠의 구조로서 역할한다. 인터넷 상의 상품정보를 모바일 폰상에서 이용하기 위한 시스템[8]은 서버/클라이언트 구조로서, 클라이언트가 상품 검색을 요청하면, 이 요청이 서버에 전달되고, 서버는 클라이언트의 요청을 해석하여, 인터넷 검색을 통하여 해당 상품 문서를 가져오고, 이 문서를 3MP 모델의 구조를 갖는 XML 문서로 변환하여 클라이언트에 전달한다. 3MP 모델은 클라이언트의 모바일 폰상에 상품정보를 효과적으로 표현하도록 설계된 XML 기반 상품표현모델이다. 클라이언트는 전달된 문서를 파싱하여 DOM 트리를 생성하고, 이 트리로부터 3MP 모델을 반영하는 상품정보 객체계층구조를 구성하고 사용자의 반응에 따라서 4단

계에 걸쳐서 상품정보를 모바일 폰의 화면상에 효과적으로 디스플레이한다.

그림 2는 모바일 클라이언트가 서버로부터 전달된 XML 상품문서를 파싱하여 DOM 트리를 생성하고, 생성된 DOM 트리로부터 상품정보 객체계층구조를 생성하고, 이 객체계층구조를 이용하여 4단계에 걸쳐서 상품정보를 디스플레이하는 과정을 보여준다. 클라이언트의 한 모듈인 Mobile Information Analyzer는 DOM 트리로부터 객체계층구조를 생성하는데, 이 구조는 그림 1의 3MP 모델을 반영하는 것을 알 수 있다. 객체계층구조에서, 상품에 대한 분류 리스트를 포함하는 SetOfCategory가 최상단에 위치하며, 각 상품 분류에 속한 상품 목록 리스트를 포함하는 CategoryOfMerchant가 다음번째 수준에 위치하며, 각 상품에 대한 세부 정보를 포함하는 Merchant 객체가 다음번째 수준에 위치한다. Merchant 객체는 해당 상품에 대한 상세 정보를 멤버로 가지며, 이러한 멤버는 변수 혹은 객체 형태로 표현된다. 구체적으로, Merchant 객체는 그림 1의 3MP 모델에서 ProductData의 구성 요소들을 멤버로 갖는다.

클라이언트의 다른 모듈 Information Viewer는 생성된 객체계층구조를 이용하여 4단계(Step1~Step4)에 걸

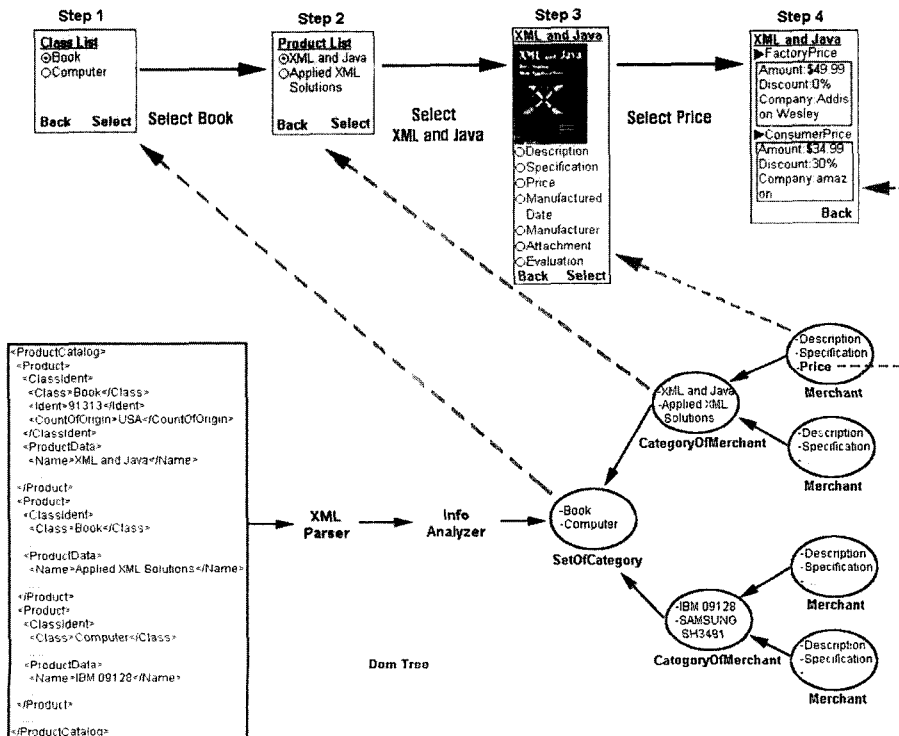


그림 2 객체계층구조를 이용한 상품정보 디스플레이 과정

처서 상품정보를 디스플레이한다. Step1에서는 객체계층구조의 SetOfCategory 객체를 이용하여 상품의 카테고리 리스트를 보여주며, Step2에서는 CategoryOfMerchant 객체를 이용하여 사용자가 선택한 상품 카테고리에 포함된 상품 목록을 디스플레이하고, Step3에서는 Step2의 화면에서 사용자가 선택한 상품에 대한 전체 정보 항목을 해당 Merchant 객체를 이용하여 디스플레이하고, 마지막으로 Step4에서는 Step3의 화면에서 사용자가 관심있는 항목을 선택했을 때, Merchant 객체의 해당 멤버를 이용하여 항목에 대한 세부 정보를 디스플레이한다.

클라이언트가 생성하는 객체계층구조 구성 방식은 논문에서 고려하고 있는 객체계층구조 구성 방식 4가지 중에서 정적 방법에 해당된다. XML 상품문서가 클 경우에도 클라이언트가 이를 효과적으로 다룰 수 있는 다른 방법들을 고려하고, 이러한 방법들을 평가 비교하고, 그 결과에 기반하여 상황에 따라 효과적으로 적용할 수 있는 방법을 추천하는 것이 논문의 목적이다.

3. 상품정보 객체계층구조 구성 방법

그림 2에서 클라이언트의 Info Analyzer가 생성하는 객체계층구조는 상품정보에 따라서 그 크기가 상당할 수 있으며, 이 경우에 그 구조를 생성하는데 상당한 시간이 소요되어 클라이언트에 대한 응답시간이 느려지는 문제점이 초래된다. 또한 생성된 객체계층구조를 저장하기 위해 상용한 크기의 기억공간도 필요하게 되는데, 현재의 기술수준에서 불 때 모바일 폰 장치의 용량 및 처리 능력이 제한되어 문제점을 야기한다. 다음은 이러한

문제점들을 해결할 수 있는 한 방안으로, 객체계층구조를 효과적으로 생성하고 이용할 수 있는 4가지 방법, 정적, 혼합, 향상된 혼합, 동적 방법을 제안하고 논의한다.

3.1 정적(static) 방법

객체 계층구조 구성에 있어서, 정적 방법은 상품정보 디스플레이의 첫 번째 단계, Step1의 상품의 카테고리 리스트를 디스플레이하기 전에, 객체계층구조 전체를 미리 구성하는 방법이다. 이 방법은 객체 계층구조 전체를 구성한 후에 Step1 화면에 대한 응답시간이 상당히 지연될 수 있지만, Step2부터는 그 구조가 이미 구성되어 있기 때문에, 매우 빠른 응답 시간을 제공할 수 있다. 따라서 첫 번째 단계와 그 이후의 단계간에 응답시간 편차가 매우 클 것이다. 또한, 객체 계층구조 전체가 생성되어 클라이언트 기억공간상에 저장되어야 한다.

3.2 동적(Dynamic) 방법

동적 방법은 Step1부터 Step3에 이르기까지 사용자의 선택에 반응하여 객체계층구조의 필요한 부분만을 동적으로 점차적으로 구성해 가는 방법이다. 그림 4는 이 과정을 보여준다. Step1 화면 디스플레이를 위해서, 단지 객체계층구조의 최상위 수준 노드 SetOfCategory만을 구성하고, 이 노드를 이용하여 Step1 화면을 구성하여 디스플레이한다. 다음에, 사용자가 Step1 화면에서 관심 있는 상품 분류를 선택하면, 해당 CategoryOfMerchant 노드만을 생성하여 객체계층구조 일부분을 구성하고, 그 노드를 이용하여 Step2 화면을 구성하여 디스플레이한다. 동일한 방법으로, 사용자가 Step2 화면에서 관심 있는 상품을 선택하면, 해당 Merchant 노드만을 생성하고, 좀더 커진 객체계층구조 일부분을 구성하고, 생성된

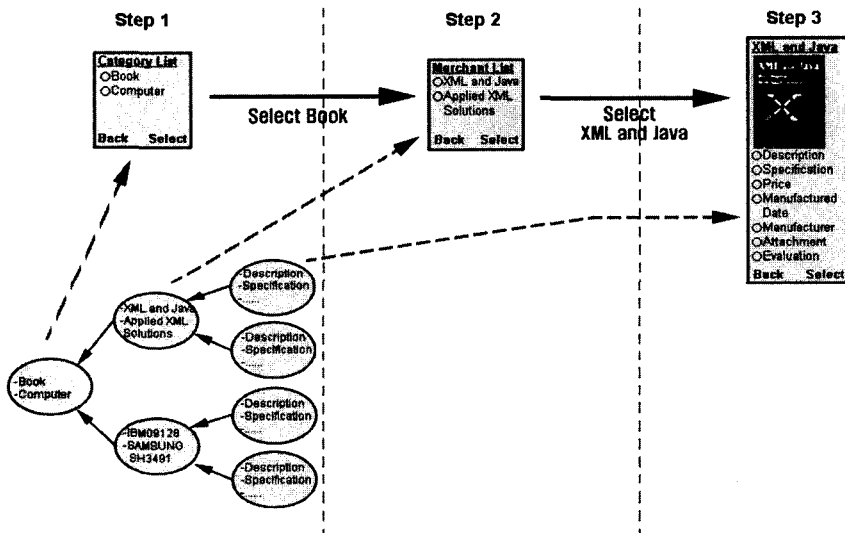


그림 3 정적 방법

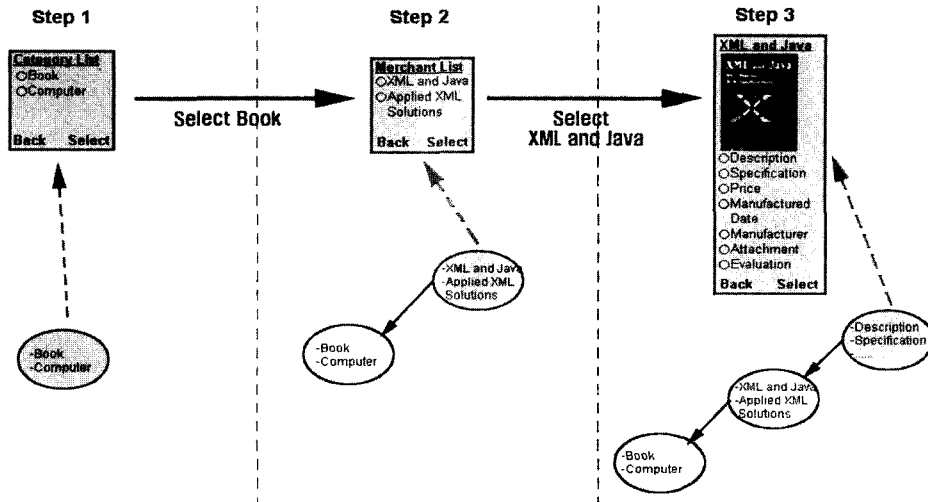


그림 4 동적 방법

Merchant 노드를 이용하여 Step3 화면을 구성한다. 이 방법은, Step1 화면을 디스플레이하는데 걸리는 시간은 정적 방법과 비교하여 상당히 단축시킬 수 있다. 그러나 Step2 이후 화면을 구성하여 디스플레이하는데 소요되는 시간은 해당 노드를 생성하여 부분 객체계층구조를 구성해야 하기 때문에, 정적 방법에 비해서 약간 더 많은 시간이 요구된다. 그러나 사용자의 응답에 반응하여 단지 필요한 노드들만으로 구성된 객체계층구조의 일부분만을 생성함으로써 기억공간을 상당히 절감할 수 있다.

3.3 혼합(Hybrid) 방법

이 방법은 동적 방법과 마찬가지로 단지 객체계층구조의 최상위 수준 노드 SetOfCategory만을 구성하고, 이 노드를 이용하여 Step1 화면을 구성하여 디스플레이한다. 그러나 사용자가 Step1 화면에서 관심 있는 상품 분류를 선택했을 때, 해당 CategoryOfMerchant 노드뿐만 아니라 그 분류에 속한 상품을 표현하는 모든 Merchant 노드들도 함께 생성하여 객체계층구조의 일부분을 구성하고, 다음에 생성한 CategoryOfMerchant 노드를 이용하여 Step2 화면을 구성하여 디스플레이한

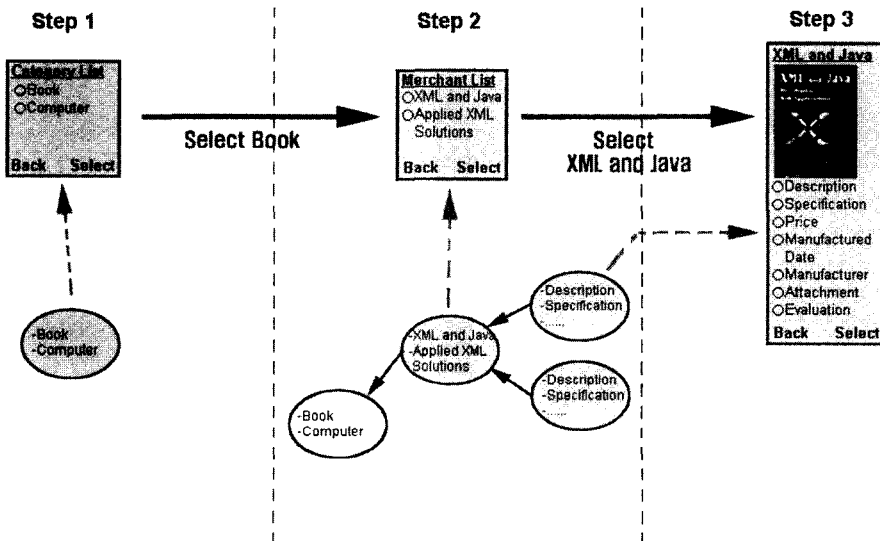


그림 5 혼합 방법

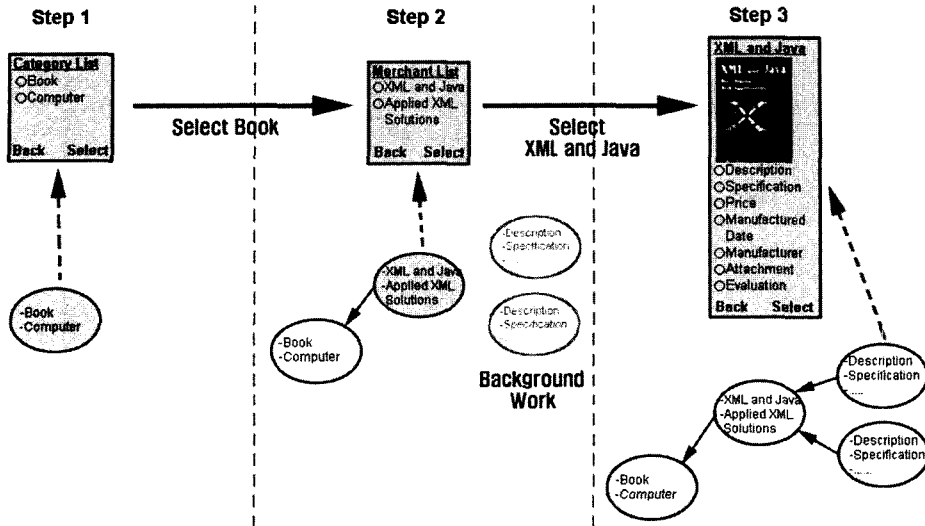


그림 6 향상된 혼합 방법

다. 따라서 Step2 화면에 대한 응답시간은 동적 방법에 비해서 늘어난다. 그러나 Step3 화면은 이미 객체계층 구조에 존재하는 Merchant 노드를 사용하여 디스플레이될 수 있기 때문에 정적 방법과 동일하게 빠른 응답 시간을 제공할 수 있다. 또한, 객체계층구조는 전체와 비교해서 약 절반정도만이 구성될 수 있기 때문에 기억 공간을 절감할 수 있다. 이 방법은 사용자가 상품 분류를 선택하면, 그 분류에 속한 상품들을 주로 검색하게 되는 일반적인 패턴을 반영한다. 즉, 사용자가 특정 분류에 속한 특정 몇몇 상품만을 검색할 경우에는 동적 방법이 뛰어나지만, 여러 상품들을 검색하게 될 경우에는 혼합 방법이 더 나을 수 있다.

3.4 향상된 혼합(Advanced Hybrid) 방법

이 방법은 결과적으로 혼합 방법과 동일하며, 차이점은 SetOfCategory 노드의 하위 부분을 구성하는 방법에 있다. 혼합 방법에서는 CategoryOfMerchant 노드뿐만 아니라 그 노드의 모든 자식 노드, Merchant 노드들로 구성된 객체계층구조를 구성한 후에 Step2 화면을 구성하여 디스플레이하기 때문에 Step2 화면에 대한 응답 시간이 어느 정도 지연된다. Step2 응답시간 지연을 개선하기 위한 방법으로, CategoryOfMerchant 노드를 생성하여 부분 객체계층구조를 구성한 즉시 Step2 화면을 구성하여 디스플레이하고, 사용자가 Step2 화면에 대해서 특정 상품을 선택하는 동안에, 후면 처리(background processing) 방식으로 그 노드의 모든 하위 노드들을 생성하여 부분 객체계층구조에 추가하는 것이다. 사용자가 Step2 화면을 보고 특정 상품을 선택하는데 어느 정도의 시간이 소요될 것이고, 사용자가 특정 상품

을 선택할 즈음에, 클라이언트가 해당 CategoryOfMerchant 노드의 모든 하위 노드들을 생성하여 부분 객체계층구조를 이미 구성했을 가능성이 많다. 따라서 Step2 화면에 대한 응답시간을 정적 방법의 수준으로 향상시킬 수 있다.

4. 실험

3장에서 살펴본 객체계층구조 구성 방법 4가지를 사용자 요청에 대한 응답 시간과 메모리 사용량의 관점에서 비교, 분석한다. 이를 위해서 [14]에서 구현된 인터넷 상품정보를 이용할 수 있는 클라이언트-서버구조의 시스템을 이용한다. 이 시스템에서 서버는 Windows 2000 운영체제를 사용하고 jakarta-tomcat 웹 서버[17]를 사용하고 있으며, 그 구성 모듈은 J2EE[18]에 기반하여 구현되었고, 클라이언트 시스템은 J2ME[19]에 기반하여 구현되었고, Wireless Toolkit for windows[20]상에 탑재되어 실행되었다. 클라이언트의 XML Parser는 트리 기반 파서인 Tiny XML 파서[21]가 사용되었다. 향상된 혼합 방법에서 후면 처리는 Java 스레드를 사용하여 구현하였다. 객체계층구조 구성 방법 비교를 위한 응답시간은 프로그램 코드 상에 삽입된 시간측정함수를 이용하여 측정되었으며, 메모리 사용량은 Toolkit이 제공하는 사용자가 정의한 단계별로 메모리 사용량을 보여주는 그래프를 이용하여 측정되었다.

실험을 위하여 책, 컴퓨터, 음반의 상품정보를 포함한 XML 문서가 작성되었으며, 그 문서의 크기는 3.21KB 이다. 실험에서 클라이언트가 이 문서를 요청하고, 서버는 이 문서를 3MP 구조를 갖는 XML 문서로 변환하여

클라이언트에 전달하고, 클라이언트는 이 문서를 파싱하여 DOM 트리를 구성하고, 이 DOM 트리로부터 객체계층구조를 구성하여 상품정보를 단계적으로 디스플레이한다.

그림 7과 8은 상품정보를 디스플레이하는 과정에서, 화면 응답시간과 메모리 사용량을 5개의 단계로 구분하여 보여준다. 5개의 단계는 t_0, t_1, t_2, t_3, t_4 로 명세된다. t_0 는 XML 문서 파싱 이후부터 상품 분류 리스트(Step1 화면)를 디스플레이하는 시간이며, t_1 은 Step1 화면에서 사용자가 특정 상품 분류를 선택한 후 상품 목록 리스트(Step2 화면)를 디스플레이하는 시간이며, t_2 는 Step2 화면에서 사용자가 선택한 상품의 정보 항목 리스트(Step3 화면)를 디스플레이하는 시간이며, t_3 는 Step2 화면으로 되돌아가서 다른 상품을 선택했을 때 Step3 화면 디스플레이하는 시간이며, t_4 는 Step1으로 되돌아가서 다른 상품 분류를 선택했을 때 Step2 화면을 디스플레이하는 시간이다.

그림 7에서 사용자 응답시간은 밀리초(milliseconds) 단위로 측정되었다. t_0 에서의 응답시간을 생각해보자. 문서 파싱이후 Step1 화면을 디스플레이하는데 정적 방법이 200 밀리초 걸린 반면에, 다른 방법은 거의 시간이 소요되지 않은 것을 알 수 있다. 이것은 정적방법이 객체계층구조 전체를 구성한 다음에 Step1 화면을 디스플레이하는 것에 반해서, 다른 방법은 단지 객체계층구조의 루트 노드만을 구성한 다음에 응답하기 때문이다. t_1 에서 혼합 방법이 다른 방법보다 약 2배정도 더 긴 응답시간을 갖는다. 이것은 혼합 방법이 해당 상품분류 노드의 전체 하위 부분을 구성한 후에 Step2 화면을 디스플레이하는 반면에, 다른 방법은 적어도 해당 상품 리스트 노드만을 구성한 즉시 응답하기 때문이다. 정적 방법에서는 이미 상품 리스트 노드가 구성되어 있음을 유의하라.

t_2 에서 동적 방법이 다른 방법보다 2배정도 더 걸린다. 이것은 동적 방법이 Step3 화면을 디스플레이하기 전에 해당 상품 노드를 구성해야 하는 것에 반해서, 다른 방법에서는 이러한 상품노드가 이미 구성되어 있기 때문이다. 향상된 혼합 방법에서, 사용자가 Step2 화면에서 특정 상품을 선택했을 때, 해당 상품노드들이 후면 처리 과정을 통해서 이미 구성되어 있음을 유의하라. t_3 에서의 응답시간은 t_2 에서의 응답시간과 동일하다. 이것은 Step3 화면으로부터 Step2의 이전 화면으로 되돌아가는 것은 Step2 화면에 대한 첫 번째 방문 경우와 동일한 상황을 낳기 때문이다. 마찬가지로 이유로, t_4 에서의 응답시간은 t_1 에서와 거의 유사하다. 전반적으로, 향상된 혼합 방법이 가장 좋은 응답시간을 가지며, 동적 방법이 두 번째로 좋은 응답시간을 갖는다. 예상하였듯이, 정적

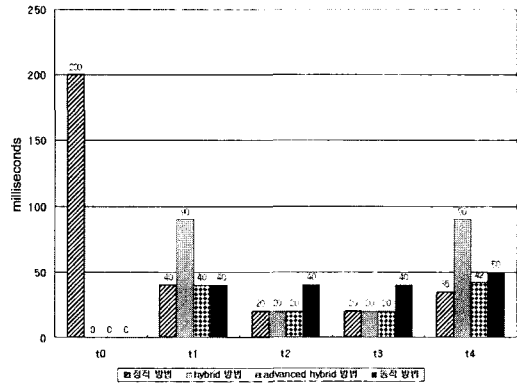


그림 7 응답 시간

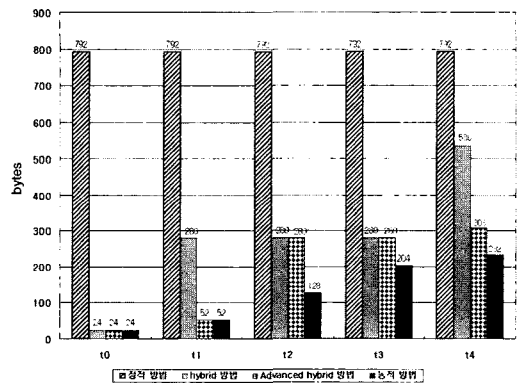


그림 8 메모리 사용량

방법이 가장 좋지 않은 응답시간을 갖는데, 이것은 사용자가 상대적으로 긴 Step1 화면에 대한 응답시간을 대기하기 어려울 수 있기 때문이다.

그림 8은 메모리 사용량 관점에서 4가지 방법을 비교한다. 정적 방법이 고정된 가장 많은 메모리를 필요로 하는 것을 알 수 있다. 이것은 정적 방법이 Step1 화면을 디스플레이하기 전에 객체계층구조 전체를 구성하고 유지해야 하기 때문이다. 반면에, 동적 방법은 응답 화면 구성에 반드시 필요한 노드만을 동적으로 구성하기 때문에 메모리 사용량이 점진적으로 증가되고 있으며, 전체 메모리 사용량도 가장 작다. 혼합 방법과 향상된 혼합 방법은 메모리 사용량이 결과적으로 동일한 것을 알 수 있다.

5. 결론

본 논문에서는 모바일 폰의 메모리 용량, 처리 능력, 디스플레이 화면 등의 제약사항을 고려하여 인터넷상의 XML 상품정보를 모바일 폰상에서 효과적으로 디스플레이하기 위한 한 방안으로, 클라이언트에서 XML 문서

로부터 객체계층구조를 구성하는 4가지 방법, 정적, 동적, 혼합, 향상된 혼합 방법을 제안하고, 실험을 통해서 4가지 방법을 사용자 응답시간과 메모리 사용량 관점에서 비교, 분석하였다. 실험 결과는 향상된 혼합 방법이 전반적으로 가장 좋은 응답시간을 보여주고, 동적 방법이 가장 작은 메모리 사용량을 보여준다. 어느 방법이 선호되어야 하는지는 XML 상품문서의 크기에 따라 다를 수 있다. 예를 들면, 필요한 메모리양이 모바일 단말기의 저장 능력을 초과할 정도로 문서가 클 경우에는 동적 방법이 사용되어야 할 것이다. 그렇지 않을 경우에는 더 나은 응답시간을 위해서 향상된 혼합 방법이 사용될 수도 있다.

향후 과제로, 논문이 제안한 방법이 인터넷 쇼핑몰에서 다루고 있는 다양한 상품정보에 대해서 실험되어야 하며, 사용자의 검색 패턴을 반영하는 것도 고려되어야 한다. 또한, 상품문서의 크기가 임의의 크기일 수 있으므로, 일반성을 위해서 문서를 적절한 크기의 세그먼트 단위로 분할하고, 사용자 요청시에 필요한 정보를 포함하고 있는 세그먼트를 클라이언트에 전달하는 방법을 연구하는 것이 필요하다. 또한, 사용자의 관심 사항만을 추출하여 필요한 정보만을 전달함으로써 모바일 클라이언트에게 전달할 데이터의 양을 크게 줄일 수 있는 적용성 있는 개인화 기술(adaptive personalization)[22]을 적용하는 것도 필요하다.

참 고 문 헌

- [1] Norman Sadeh, "M-Commerce: Technologies, Services, and Business Models," Reding, Wiley, 2002.
- [2] NTT Docomo: <http://www.nttdocomo.com>
- [3] Nordea's WAP Solo Mobile Banking Service: <http://www.nordea.com>
- [4] Webraska's SmartZone Platform: <http://www.webraska.com/>
- [5] SK텔레콤의 네이트: <http://www.nate.com>
- [6] KTF의 매직엔[5]: <http://masicN.co.kr>
- [7] LG텔레콤의 이지아이: <http://ez-i.co.kr>
- [8] AvantGo: <http://www.avantgo.com>
- [9] WebViewer: <http://www.reqwireless.com/webviewer.html>
- [10] Jones M, et al., "A site based outliner for small screen Web access," World Wide Web conference, 1999.
- [11] Bill N. Schilit, et al., "Web Interaction Using Very Small Internet Devices," IEEE Computers, Oct. 2002.
- [12] George Buchanan, et al., "Improving Mobile Internet Usability," World Wide Web conference, 2001.
- [13] IBM Transcoding solution and service: <http://www-3.ibm.com/software/webservers/transcoding/publications/transcoding.html>
- [14] 최정익, 하상호, "모바일 폰에서 인터넷 상품 정보의 효과적인 디스플레이", 정보과학회 춘계학술대회, 2003.
- [15] Html2Xml WebService: <http://www.reflectionit.nl/Html2Xml.aspx>
- [16] 하상호, 김경래, "XML 기반 상품 카탈로그의 설계 및 적용", 정보처리학회논문지, 제9-D권, 제3호, 2002년 6월.
- [17] jakarta-tomcat 웹 서버: <http://xml.apache.org>
- [18] J2EE: <http://java.sun.com/j2me>
- [19] J2ME: <http://java.sun.com/j2me>
- [20] J2ME Wireless Toolkit: <http://java.sun.com/products/j2mewtoolkit/>
- [21] TinyXML Parser: <http://www.gibardunn.srac.org/tiny/index.shtml/>
- [22] Daniel Bilsus, et al., Adaptive Interfaces for Ubiquitous Web Access, CACM, Vol. 45, No. 5, 2002.



하 상 호

1998년 서울대학교 계산통계학과(학사)
1991년 서울대학교 계산통계학과(석사)
1995년 서울대학교 전산학과(박사). 1995년~1996년 한국전자통신연구원 박사후 연구원. 1996년~1997년 미국 MIT LCS 박사후 연구원. 1997년~현재 순천향대학교 정보기술공학부 부교수. 관심분야는 프로그래밍 언어, 유비쿼터스 컴퓨팅, XML, e-commerce, 병렬 및 분산처리