

모바일 클라이언트를 위한 웹 정보 필터링¹⁾

허정수, 하상호

순천향 대학교 컴퓨터 학부

e-mail:{majya,hsh}@sch.ac.kr

Filtering Web Information for Mobile Clients

Hur Jeongsoo, Sangho Ha

Dept of Computer Science and Engineering,

Soonchunhyang University

요 약

오늘날 인터넷의 광범위한 보급은 무선 이동 통신 기술로 급발전하고 무선 단말기의 보급과 폭발적인 증가로 인해 m-commerce가 차세대 상거래의 형태가 될 것으로 예상된다. 현재 m-commerce의 콘텐츠는 e-commerce와는 호환되지 않으며, 질적으로나 양적으로 그 내용이 매우 빈약하다. 따라서 e-commerce의 상품정보 콘텐츠를 m-commerce를 위해서 활용하는 방안이 요구되는데, 여기에는 무선 단말기의 제한된 처리 능력이 고려되어야 한다. 논문에서는 먼저 인터넷 상의 방대한 상품정보를 사용자의 선호도에 기반하여 효과적으로 여과시켜서 사용자가 관심있는 정보만을 간추려서 제공할 수 있는 시스템을 설계하고 구현한다. 다음에 구현된 시스템을 인터넷 쇼핑물에 대해서 적용한다.

1. 서론

오늘날 인터넷의 광범위한 보급은 이제 유선을 넘어 무선 이동 통신 기술로의 급발전을 가져오게 되었고, 무선 단말기의 보급과 폭발적인 증가로 인해 m-commerce[1]가 차세대 상거래의 형태가 될 것으로 예상된다. M-commerce는 무선 인터넷 통신을 사용하여 이루어지는 상거래로서, 무선 인터넷 기술의 발달에 힘입어 성장으로 NTT DoCoMo의 i-Mode 포털[2], SK텔레콤의 네이트[3], KTF의 매직엔[4] 등과 같은 국내외의 많은 업체들이 m-commerce를 위한 서비스를 제공하고 있지만 e-commerce와 같이 풍부한 상품 정보가 서비스되지 않는 실정이다. 그 이유는 서비스가 단순한 수준이며, 이러한 서비스조차도 기종의 차이로 업체별로 제작되며 서로 호환되지 않기 때문이다.

논문에서는 e-commerce를 위해서 서비스되고 있는 웹상의 풍부한 상품정보를 m-commerce에 활용하는 방안을 고려한다. 현재의 m-commerce는 대부분의 사람이 휴대하고 있는 모바일 폰을 통해서 이루어지고 있으며, 모바일 폰이 앞으로 m-commerce를 위한 주요 모바일 장치가 될 것으로 예상된다. 무선 통신의 자료 전송 능력과 모바일 폰의 자료 처리 능력이 유선 컴퓨팅 환경에 비해서 상당히 떨어

지므로, 웹의 방대한 상품정보를 효과적으로 모바일 폰 사용자에게 제공하는 것이 중요하다. 논문에서는 웹의 상품정보를 사용자 선호도에 기반하여 효과적으로 여과시켜서 모바일 단말기에 전달할 수 있는 시스템을 개발한다.

관련 연구로, [5,6,7,8] 등을 들 수 있다. [5]에서는 HTML 문서를 DOM을 이용하여 PDA나 모바일 폰의 웹브라우저에서 디스플레이하기에 적절한 HTML문서로 변환한다. 사용자에게 불필요한 정보를 제외하고 필요한 정보만을 제공하나 정보의 범위가 하나의 웹 문서로 제한되며 논문에서 고려하는 상품 정보 제공에 대하여는 고려하지 않는다. [6]에서는 웹문서의 단어 중에서 사용자가 원하는 검색 문장의 동의어, 상위어, 하위어를 계산하여 사용자가 원하는 정보에 근접하는 웹문서 안의 내용을 웹브라우저 에이전트를 이용하여 사용자에게 제공한다. 그러나 사용자가 제공한 검색어 단어의 개수에 따라 정확도가 달라지며, [5]에서와 같이 하나의 웹문서만을 고려하며 상품 정보에 대하여 고려하지 않는다. [7]에서는 사용자가 관심 있는 웹 페이지 접근 과정을 Smart Bookmark로 기록하고, 해당 웹 페이지에서 추출하고자 하는 항목들을 XPath로 기술함으로써 개인화된 웹 뷰를 생성한다. 여러 웹문서의 XPath를 정의하여 한 번에 여러 웹문서에서 사용자가 원하는 정보만을 열람할 수 있으나 개인화에 대한 고려가 없으며, 사용자가 XPath를 이용하여 직접

1) 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음 (R05-2004-000-12565-0)

경로를 기술해야 하기 때문에 사용이 용이하지 않다. [8]에서는 인터넷 쇼핑몰의 HTML 상품 정보 웹 문서에서 래퍼에 기반하여 필수적인 상품 항목만을 추출한다. 그러나 이 연구는 사용자 선호도에 기반한 상품 정보 여파를 고려하지 않는다.

2장에서는 논문에서 고려하는 웹 정보 필터링 시스템에 대해서 UML에 기반하여 설계하고, 3장에서는 이 시스템에 대한 구현 사항을 기술한다. 4장에서는 구현된 시스템을 인터넷 쇼핑몰에 대해서 적용하여 마지막으로, 5장에서는 결론을 언급한다.

2. 설계

그림 1은 웹 정보 필터링 시스템의 전반적인 구조를 보여준다. 웹 정보 필터링 시스템은 필터 관리자에 의해 외부 통신이 가능하다. 우선 Filter Manager는 사용자가 필터들에 대한 정보를 요청하면 필터 설명 문서를 Filter Script에 요청하여 제공한다.

File Manager는 항상 필터가 준비 되어 있는 디렉토리를 검사하여 새로운 필터가 있거나 필터가 수정되면 입력, 수정, 삭제 등을 인식하여 Filter Script와 Filter Pool의 이벤트를 발생시켜 적용한다. 필터 설명 문서를 본 사용자는 사용할 필터를 정하고 Filter Manager에 선택한 필터들로 구성된 Filtering Engine을 요청한다. Filter Manager는 필터링 엔진 요청을 받아들여 알맞은 Filter Pool에 있는 필터를 찾아 Filtering Engine을 구성하여 사용자에게 제공한다.

그림 2는 필터링 시스템의 UML 객체 다이어그램을 나타낸다. 필터링 시스템은 FilterManager를 중심 객체로 하여 디렉토리의 필터들을 관리를 하는 FilterRegister와 필터를 적재하고 관리하는 FilterPool이 있다. 또한 FilterManager에 의해 생성되어 실제 필터링 작업을 수행하는 FilterEngine이 있다.

FilterManager는 FilterPool과 FilterRegister를 갖고, setFilterPool과 createFiltering 메서드가 존재한다. setFilterPool은 FilterPool을 받아 FilterRegister에 등록한다. 이로써 FilterRegister는 FilterPool에 이벤트를 발생시켜 필터를 등록 및 삭제, 수정할 수 있다. 또한 createFiltering은 필터링 조건인 Condition의 리스트를 받아들여 FilterPool에서 Condition에 부합하는 필터를 선별하여 FilterEngine을 구성한다.

2.1 IFilter와 Filter Script

필터링을 위한 인터페이스와 그에 대한 XML 기술 문서를 말한다. IFilter를 상속하여 구현된 필터는 본 논문의 필터링 시스템에서 사용할 수 있다. 그런데 IFilter를 상속하여 구현된 객체만 이용하는 것은 사용자의 의사가 적용되는 시스템에는 알맞지 않다. 그래서 본 논문에서는 이 필터의 사용에 대한 기술 문서인 Filter Script를 필요로 한다.

Filter Script의 형태는 그림 3와 같다. Filter Script는 XML 문서 구조이며, Filters를 루트로 하여 하나 이상의 Filter를 갖을 수 있다. Filter는 필터의 이름을 나타내는 name속성, 입력 받는 방법에 대한 type속성, 사용자에게 보이기 위한 설명인 text

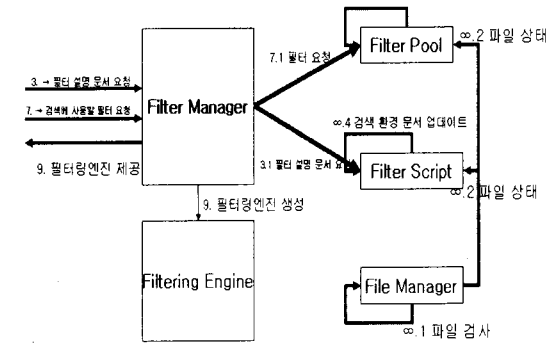


그림 1 시스템 구조

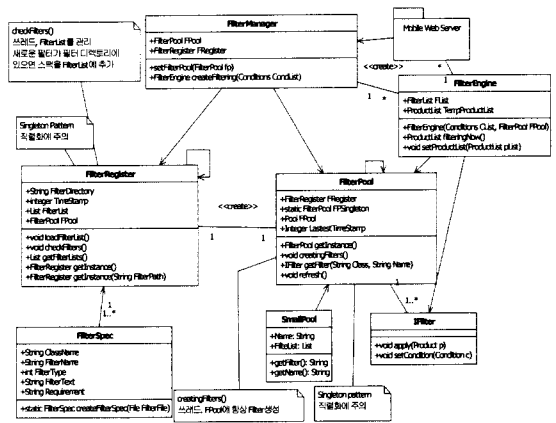


그림 2 필터링 시스템의 UML 클래스 다이어그램

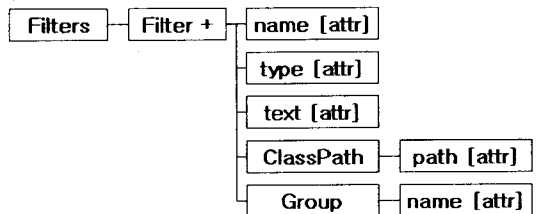


그림 3 Filter Script의 구조

속성을 갖고 있다. 또한 Filter는 IFilter를 상속해 구현된 필터 클래스의 정의를 위한 ClassPath가 자식 노드로 존재하며, ClassPath는 path속성으로 필터의 클래스 파일의 이름을 저장한다. 마지막으로 Group은 구현된 필터의 구분을 위한 것으로 name속성으로 현재의 Filter Script가 지정하고 있는 필터에 대한 그룹을 알 수 있다.

2.2 FilterRegister

필터가 준비되어 있는 디렉토리를 주기적으로 검사하여 새로운 필터나 변경, 제거된 필터에 대한 정

보를 업데이트하는 역할을 한다. FilterRegister는 디렉토리를 검사하여 새로운 필터가 있으면 2.2절에서 언급한 Filter Script를 읽어 FilterSpec클래스를 생성한다. FilterSpec클래스는 Filter Script의 매핑 클래스이며, Filter Information 문서에 저장하고 FilterPool에 FilterSpec을 제공하여 필터 클래스를 적재할 수 있게 한다.

2.3 Filter Information

필터를 사용하는 사용자를 위하여 필터에 대한 정보가 저장되는 Filter Information문서를 생성하는 역할을 한다. FilterManager에서 이벤트가 발행하면 FilterSpec을 받아 Filter Information 문서에 FilterSpec의 내용을 저장하여 사용자로 하여금 필터를 사용할 수 있는 방법을 제공한다.

Filter Information은 그림 4과 같다. Filter Information은 XML 구조이며, Script를 루트로 하고 필터들의 집합인 Group이 있다. Group은 Name속성으로 그룹의 이름을 정할 수 있으며, 내부에 여러 Filter를 정의할 수 있다. Filter는 Name속성과 필터의 사용 방법을 나타내는 Type속성, 필터에 대한 설명인 내부 문자열(#Text)를 갖는다.

Filter Information에 FilterSpec의 추가가 끝나면 FilterManager는 FilterPool에 필터에 대한 정보를 알린다. 이것은 IPool인터페이스에 의해 정의된 메시지를 이용하여 이벤트를 호출하듯이 새로운 필터는 insert 메시지, 필터 정보 변경은 update, 필터 삭제는 remove 메시지를 실행하며 매개 변수로는 FilterSpec을 제공한다.

2.4 FilterPool

FilterRegister에서 필터에 대한 정보를 메시지 별

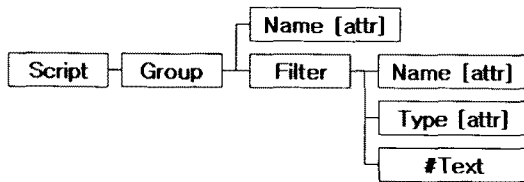


그림 4 Filter Information의 구조

로 받아들여 실제 구현된 클래스들에 대해 관리하는 역할을 한다. 각 Filter에 따라 FilterSpec에 정의된 ClassName을 이용하여 객체를 적재하고 SmallPool에 저장한다.

SmallPool은 Pool패턴을 이용한 객체로서, 필터 클래스를 적재하여 일정한 수의 필터들을 미리 생성하여 사용자가 원할 때 제공하는 역할을 한다. SmallPool은 FilterPool에 필터 하나 당 하나의 SmallPool로 구성되며, 필터 이름에 의해 구분된다.

FilterPool은 FilterSpec을 이용하여 신규 필터 등록, 기존 필터 업데이트, 기존 필터 제거의 작업을 한다. 여기서 신규 필터 등록은 FilterSpec에 있는 필터의 이름인 FilterName과 구현된 필터의 위치를 나타내는 ClassName을 사용하여 SmallPool을 생성하고 SmallPool들을 관리하며, 사용자가 원하는 필터를 구해서 제공한다.

2.5 FilterEngine

시스템이 정상적으로 구동되는 가운데, 사용자의 필터링 요청 시에 FilterManager에서 사용자가 원하는 필터만을 FilterPool에서 찾아 FilterEngine을 구성하여 사용자에게 제공된다. FilterEngine은 FilterManager에서 받은 사용자 시스템은 필터링을 위한 정보를 넣으면서 FilterEngine의 loadFiltering 메시지를 실행하여 필터링된 정보를 얻을 수 있다.

3. 구현

본 논문의 필터링 시스템은 자바 플랫폼에서 구현되었다. 필터링 시스템은 하나의 컴포넌트임으로 다른 시스템과 독립성을 높여, 정보를 수집하는 시스템에서 필터링 모듈로 활용할 수 있는 방법을 제공한다. 그러나 필터링 시스템만 사용할 수 없으므로 선행 연구된 상품 정보 추출기에 적용하였다. 상품 정보 추출기는 웹 정보 추출[8]을 이용하여 인터넷 쇼핑몰의 상품의 정보를 추출하여 상품 정보를 만들어내는 시스템으로 단어에 의해 이루어진 검색 결과에 필터링 시스템을 포함하여 사용자 개인화를 실현하였다. 필터는 상품명 상세 검색 필터와 제조사명 상세 검색 필터, 두개의 필터를 구현하였으며 필터링 대상은 상품 정보에 제한하여 실행된다.

4. 적용

본 논문에서 제시한 필터링 시스템이 적용된 상품 정보 추출 시스템을 이용하여 적용 예를 보인다. 상품 정보 추출 시스템에 본 필터링 시스템은 하나의 모듈로 포함이 되며 필터링의 대상은 인터넷 쇼핑몰 중 인터넷파크[9]와 GS이숍[10]에서 추출된 상품 정보에 대하여 고려한다.

상품 정보 추출 시스템이 실행되면 FilterManager

```

<?xml version="1.0" encoding="utf-8" ?>
<Filters>
  <Filter name="Company_Filter"
    type="String"
    text="제조사는 다음 단어를 포함">
    <ClassPath
      path="CompanyFilter.class" />
    <Group name="Default" />
  </Filter>
</Filters>
    
```

그림 5 Company 필터 정의 문서

```

<?xml version="1.0" encoding="EUC-KR"?>
<Script>
  <Group Name="Default">
  <Filter Name="Company_Filter" Type="String">제조사는 다음 단어를 포함</Filter>
  <Filter Name="Title_Filter" Type="String">상품명은 다음 단어를 포함</Filter>
  </Group>
</Script>
    
```

그림 6 Filter Information 문서

가 만들어지고 FilterManager는 FilterRegister를 통해 필터들이 모인 디렉토리를 검색한다. 검색하여 새로운 필터가 생기면 필터 정의 문서를 이용하여 FilterSpec을 생성함으로써 3장에서 밝혔듯이 필터 2개를 구현하였다. 그림 5는 두 개의 필터 중 하나의 필터의 필터 정의 문서를 보인다.

FilterSpec이 생성되면 Filter Information 문서가 만들어진다. 그림 6는 필터 2개가 등록되어 있는 Filter Information 문서를 보인다. 그리고 필터 스펙은 FilterPool로 보내져 SmallPool을 생성하고 사용자의 필터 사용을 기다린다.

사용자는 Filter Information을 이용하여 적절한 필터를 선택하여 검색한다. 이 적용에서는 상품 검색에 Company_Filter를 사용하여 제조사는 "삼보"라는 필터링을 요청하면 그림 7과 같은 결과가 나온다. 참고로 그림 8은 필터링을 사용하지 않을 때의 결과물을 보인다. 그림 8의 여과기를 적용하지 않은 결과에서는 인터파크와 GS이숍에서 컴퓨터라는 검색어를 이용하여 검색한 결과가 담겨있으나 사용자가 제조사에 대하여 여과를 요청함으로써 그림7에서는 제조사명에 "삼보"라는 단어가 들어가 있는 상품들만 검색되어 표현된다.

5. 결론

본 논문에서는 인터넷에서 추출된 정보 중에서 사용자가 원하는 정보만을 이용할 수 있는 여과 시스템을 제안하였으며, 설계 및 구현을 보았다. 여과 시스템은 컴포넌트 기반으로 구성되었으며, 별도의 다른 필터에 대하여 정의하고 시스템에 쉽게 적용하여 사용자에게 필터 서비스를 제공할 수 있다. 그래서 정보 필터를 위하여 시스템의 변경이 불필요하며, 본 보고서에서 사용되는 상품 정보 추출 시스템 이외의 다른 정보 수집 시스템에도 적용 가능하다.

본 논문에서 제안하고 개발된 시스템 원형은 아직 많은 개선이 필요하다. 첫째로 필터링 엔진에서 필터들이 구성되어 정보를 여과할 때, 여과 작업이 순

```
<?xml version="1.0" encoding="EUC-KR"?>
<ItemList>
<Item>
<Product_Title>드림컴퓨터...DVD-RW </Product_Title>
<Company Alias="제조사/원산지">드림컴퓨터 </Company>
<Price Alias="판매가">645,000원</Price>
<Shop Address="www.interpark.com">인터파크</Shop>
</Item>
<Item>
<Product_Title>[2005년 맞춤PC]... </Product_Title>
<Company Alias="제조사/원산지">하드피아 </Company>
<Price Alias="판매가">120,700원</Price>
<Shop Address="www.interpark.com">인터파크</Shop>
</item>
...
</ItemList>
```

그림 8 여과기를 적용하지 않은 결과

2002

- [2] NTT DoCoMo : <http://nttdocomo.com>
- [3] SK 텔레콤 네이트 : <http://www.nate.com>
- [4] KTF의 매직엔 : <http://magicn.co.kr>
- [5] Suhit Gupta 외 3, DOM-based Content Extraction of HTML Documents
- [6] Kwang Mong Sim, Web Agents with a tree-stage Information Filtering Approach
- [7] Mehmet Altinel, Michael J. Franklin, Efficient Filtering of XML Documents for Selective Dissemination of Information
- [8] Jungsu Hur, Sangho Ha, "Packaging Web Product Information in XML for Mobile Clients", IEEE, PRIWEC 2006
- [9] 인터파크 <http://www.interpark.com>
- [10] GS이숍 <http://www.gshop.co.kr>

```
<?xml version="1.0" encoding="EUC-KR"?>
<ItemList>
<Item>
<Product_Title>[20,000원 ... 지 </Product_Title>
<Company Alias="제조사/원산지">삼보 </Company>
<Price Alias="판매가">1,096,350원</Price>
<Shop Address="...">인터파크</Shop>
</item>
<Item>
<Product_Title>울인원PC! 루온</Product_Title>
<Company Alias="제조사/원산지">삼보컴퓨터/한국 </Company>
<Price Alias="판매가">1,799,000원</Price>
<Shop Address="...">GS이숍</Shop>
</Item>
...
</ItemList>
```

그림 7 여과기를 적용한 결과

차적으로 이루어지므로 많은 데이터를 처리 시 많은 비용을 필요하다. 한 가지 개선 방안은 다중 스템딩 방식을 이용하여 멀티 필터링 기능을 고려하는 것이다. 두 번째로 다른 필터의 추가를 위하여 사용되는 필터 정의 문서에서 인터넷 상의 다양한 데이터형의 일부만을 지원하고 있다. 이는 점차 지원하는 데이터형을 늘려갈 것이다. 마지막으로 여과 기능을 하는 필터가 부족하다. 부족한 필터를 보충하기 위하여 정의한 인터페이스를 이용, 다양한 필터를 추가할 것이며, 앞으로 사용자 프로파일을 이용하여 사용자 선호도를 반영할 수 있는 방안이 고려된다.

[참고 문헌]

[1] Norman Sadeh, M-Commerce: Technologies, Services, and Business Models, Reading, Wiley,