XML기반 가상문서에서의 멀티미디어 및 구조적 문서의 표현과 처리

박천수°, 임동수, 박종현, 강민구, 강지훈
충남대학교 컴퓨터과학과
{bettle, lastdkht, jhpark, mgkang, jhkang}@cs.cnu.ac.kr

Representing and Processing Multimedia and Structured Documents For XML-Based Virtual Documents

Cheon-Shu Park°, Dong-Soo Lim, Jong-Hyun Park, Min-Gu Kang, Ji-Hoon Kang
Dept. of Computer Science, Chungnam National University

요 약

가상문서는 웹 상에 존재하는 내용 중에서 원하는 부분만을 링크를 이용해 새로운 문서를 생성하는 개념이다. 본 논문에서는 가상문서를 지원하는 디지털 도서관 시스템에서 텍스트, 이미지 데이터 뿐 아니라 멀티미디어 데이터와 구조적 의미를 갖는 데이터를 처리 가능 하도록 DTD의 표기법을 확장하였다. 또한, 저작도구에서 생성된 내포링크, 참조링크, 총칭링크 등 다양한 의미의 가상문서를 브라우징 가능하도록 문서 변환기에서 멀티미디어와 구조적 문서를 처리하기 위한 방법을 제시하였다.

1. 서론

웹상에는 수없이 많은 데이터가 존재한다. 수 많은 데이터 중에는 텍스트 형태의 데이터 뿐만 아니라 이미지, 오디오, 비디오와 같은 다양한 종류의 미디어 데이터가 존재한다. 이렇게 이미 존재하는 수 많은 미디어 데이터 중에서 자신이 필요한 부분만을 가져와 새로운 문서로 만들어야 하는 경우가 종종 있다. 이러한 요구를 충족 시키기 위한 새로운 개념의 문서가 가상문서이다.[Mya99].

가상 문서는 인터넷 상의 문서를 일부 또는 전부를 링크로 연결함으로써 새로운 문서를 생성할 수 있고, 문서에 대한 링크만 가지고 있어서 중복된 문서저장으로 인한 저장공간의 낭비를 줄일 수 있으며, 문서의 부분만을 필요로 할 경우 문서의 전체가 아닌 일부분만 전달되기 때문에 네트워크의 통신부하를 감소시킬 수 있다 [Mya99]. 이러한 가상문서를 디지털 도서관 시스템이 지원하기 위해서는 정형화된 가상문서 양식이 필요하며, 표현력, 처리 효율성, 인터넷 접근성 등을 고려할 때 XML[XML98] 이 적합하다고 판단하였으며, 이에 따라 가상문서의 양식과 관련 스타일쉬트 양식이 우리의 선행 연구를 통하여 XML DTD 로 정의되었다 [Mya99, 강99].

* 이 연구는 충남대학교 소프트웨어연구센터의 재정지원을 받았음.

본 논문에서는, 텍스트 및 이미지만을 지원하는 기존의 가상문서[Mya99, 강99, Mya00] 가, 비디오와 오디오등 멀티미디어와 XML, HTML 등 구조적 문서를 지원할 수 있는 방안을 제시한다. 가상문서에서 멀티미디어와 구조적 문서의 표현을 위하여 기존의 표현 방법의 확장을 논하며, 또한 이와 같이 표현력이 확장된 가상문서를 인터넷 상에서 브라우징할 수 있도록 가상문서를 변환 처리하는 방법을 제시한다.

논문의 구성은 다음과 같다. 2 장에서는 가상문서의 개념과 가상문서를 위한 XML DTD 를 설명한다. 3 장에서는 멀티미디어와 구조적 문서의 표현과 그 처리 방법을 설명한다. 4장에서는 관련연구를 살펴보고, 5 장에서 결론 및 향후 연구 방향에 대하여 논한다.

2. 가상문서

2.1 가상문서의 개념

가상문서는 분산환경에 이미 존재하는 문서(물리적문서)에서 연관성이 있는 내용을 링크만을 이용해 표현한 새로운 문서이다. 즉, 새로 만드는 가상문서에는 실제 내용은 존재하지는 않고 기존에 존재하던 문서로의 링크만을 갖게 된다. 가상문서는 허브(hub)와 스타일시트(stylesheet)로 구성된다. 허브는 문서의 구조와 내용을 나타내고, 스타일시트는 가상문서 전체 및 구성요소를 위한 스타일 정보를 가진다[Mya99, 강99].

2.2 가상문서를 위한 DTD

XML로 가상문서를 표현하기 위해서는 가상문서의 특성에 맞게 문서의 형식을 정의해야 한다. [그림 1] 은 가상문서의 DTD이다.

```
<!ELEMENT   VdocHub      (ELinkSeq, RLinkSet, Metadata)          >

<!ELEMENT   ELinkSeq     (ELink)+ >
<!ELEMENT   ELink        EMPTY >
<!ATTLIST   Elink
            href         CDATA                  #REQUIRED
            role         CDATA                  #IMPLIED
            title        CDATA                  #IMPLIED
            owner        CDATA                  #IMPLIED
            date         CDATA                   #IMPLIED
            category     CDATA                   #IMPLIED
            actuatedefault              (user|auto)    "user"
            autoDelete                  (NO|YES)       "NO" >

<!ELEMENT   RLinkSet                    (RLink*) >
<!ELEMENT   RLink                       (Source, Destination+)      >
<!ATTLIST    RLink
            showdefault                 (new|parsed|replace) "replace"
            actuatedefault              (user|auto)          "user  >
<!ELEMENT   Source                      EMPTY                       >
<!ATTLIST   Source
            is_generic   (NO|YES)                   "NO"
            href         CDATA                      #REQUIRED
            role         CDATA                      #IMPLIED
            title        CDATA                       #IMPLIED
            autoDelete   (NO|YES)                   "NO"        >
<!ELEMENT   Destination                 EMPTY                       >
<!ATTLIST   Destination
            href         CDATA                      #REQUIRED
            role         CDATA                      #IMPLIED
            title        CDATA                      #IMPLIED
            owner        CDATA                      #IMPLIED
            date         CDATA                       #IMPLIED
            category     CDATA                       #IMPLIED
            autoDelete   (NO|YES)                   "NO"        >

<!ELEMENT   Metadata
      (DC_TITLE?,DC_CREATOR?,DC_SUBJECT?,DC_DESCRIPTION?,
      DC_PUBLISHER?,DC_CONTRIBUTOR?,DC_TYPE?,DC_DATE?,
      DC_FORMAT?,DC_IDENTIFIER?,DC_SOURCE?,DC_LANGUAGE?,
      DC_RELATION?, DC_COVERAGE?, DC_RIGHTS?)                   >

<!ELEMENT   DC_TITLE     (#PCDATA)                              >
<!ATTLIST   DC_TITLE
            value        CDATA                      #IMPLIED    >
                              .
                              .
                              .
<!ELEMENT                DC_RIGHTS                  EMPTY       >
<!ATTLIST                DC_RIGHTS
            value        CDATA                      #IMPLIED    >
```

[그림 1] 가상문서를 위한 DTD.

가상문서는 허브와 스타일시트로 구성된다. 허브는, 내포링크 순서 (a sequence of embedding links), 참조링크 집합 (a set of referential links), 그리고 메타데이타 (metadata) 의 세 개로 구성된다. 즉, VdocHub 는 가상문서에서 가장 상위 요소 (Element)로 ELinkSeq, RLinkSet, Metadata 의 순서를 갖는 세 개의의 요소를 가지고 있다.

내포링크는 가상문서가 실체화 될 때 그 구성 요소가 되는 문서들에 대한 링크를 말한다. 가상문서 상태에서는 단지 링크이지만, 클라이언트가 가상문서를 브라우징하게 되면 그 가상문서가 가지고 있는 모든 내포링크의 목적지에 존재하는 문서들이 불려와서 하나로 취합이 되어 온전한 문서 형태로 클라이언트에게 보여야 한다. 참조링크는 링크의 다양한 방향을 지원한다. 기존의 HTML 문서에서 사용하던 단방향 링크는 물론, 양방향링크 (bi -directional link), 목적지가 여러 곳인 다중링크 (multi -directional)를 지원한다.

3. 가상문서에서의 멀티미디어와 구조적 문서

3.1 멀티미디어와 구조적 문서의 표현

미디어 데이터의 종류에 따라서 각각의 표기방법을 결정한다. 텍스트 데이터의 표기 방법은 #text 라는 한정어를 두어 Start, Offset 으로 부분문서를 표현한다. Start 는 부분문서가 시작되는 byte 이고, Offset 은 포함될 데이터의 범위를 지정한다. <ELink href="dl.cnu.ac.kr/pdoc/bach.txt#text(Start, Offset)" ...

이미지 데이터는 #image 라는 한정어를 두고 x1, y1, x2, y2 의 좌표 값을 이용해 표현 한다.
<ELink href="dl.cnu.ac.kr/pdoc/bach.gif#image(x1, y1, x2, y2)" ...

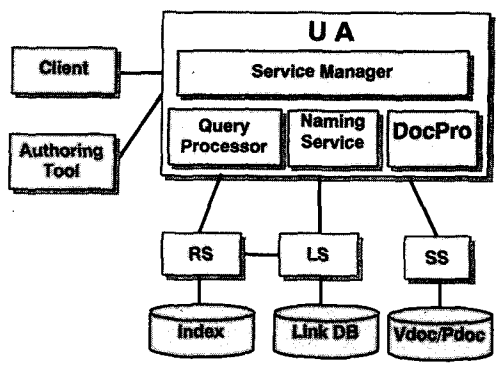
오디오 데이터는 #audio 라는 한정어를 두고 Start 와 O ffset을 이용해 부분문서를 표현한다. Start 는 부분 데이터가 검사된 시각(초)을 나타내고, Offset 은 재생되는 동안의 시각(초)을 나타낸다.
<ELink href="dl.cnu.ac.kr/pdoc/bach.mp3#audio(Start, Offset)" ...

비디오 데이터는 #video 라는 한정어를 두어 Start 와 O ffset을 이용해 부분문서를 표현한다. Start 는 데이터가 검사된 시각(초)을 의미하고, Offset은 재생 시각을 나타낸다.
<ELink href="dl.cnu.ac.kr/pdoc/bach.mpeg#video(Start, Offset)" ...

구조적 의미를 갖는 데이터(HTML, XML) 는 #struct 라는 한정어를 두고 Root 노드를 기준으로 child/grandchild/ … 형태의 ' /' 를 이용해 구조적 데이터의 위치 정보와 범위를 표현 한다.
<ELink href="dl.cnu.ac.kr/pdoc/bach.html#struct(1/2/3, 1/2/5)" ...

3.2 시스템 구조

우리의 디지털 도서관 시스템[Mya99]에서 문서처리 모듈 (Document Processing Module) 인 DocPro(문서 처리기)는 UA(User Agent) 의 한 부분으로 가상문서의 처리를 담당한다. 문서처리기는 가상문서의 고유 ID(Ide ntifier) 생성과, 링크서비스에 필요한 링크정보를 반환하며, 문서의 검색을 위한 가상문서 ID 와 문서에 정의된 메타데이타, 내포 문서들과 그 문서에 대한 ID 들을 반환해준다. 그리고 주기능인 문서변환의 기능을 가지고 있다. [그림 2] 은 디지털 도서관시스템에서 가상문서 처리를 위한 시스템의 구조를 나타내고 있다.



[그림 2] 가상문서 처리를 위한 시스템의 구조.

3.3 멀티미디어와 구조적 문서의 처리

문서 처리기에서 가상문서를 사용자가 볼 수 있도록 변환하는 가장 핵심적인 역할은 문서 변환기가 담당한다. 문서 변환기는 가상문서를 Well-formed XML 문서로 변환한다. 문서변환기는 가상문서와 스타일 문서를 받아들여 XML 파서를 이용해 트리 형태의 DOM [DOM98] 표현을 만든다. DOM 트리로부터 변환에 필요한 정보를 추출 한다. 추출된 정보에서 내포링크를 순차적으로 처리하여 문서를 가져와서 결합시켜 하나의 문서를 만든다. 여기에 참조링크와 총칭링크를 추가한다. 또한, 가상문서의 스타일 정보를 이용하여 변환된 문서를 위한 XSL[XSL99] 스타일쉬트를 만든다. [강99, Mya00]

기존에 문서처리기는 텍스트와 이미지 데이터만을 처리할 수 있도록 설계하였다. 이에 문서 처리기에서 멀티미디어 콘텐츠를 XML 형태로 변환 할 수 있도록 SS 에서 오디오, 비디오, 구조적 의미를 갖는 데이터를 처리 하여 문서 처리기에 전달 한다. 오디오, 비디오와 같이 시간적인 정보를 포함하고 있는 데이터는 전체 데이터를 요구 할 경우와는 달리 부분적인 문서를 요구할 경우 해당 부분에 대한 인코딩(encoding)이 필요하다. 이에 자바 기반의 멀티미디어 프로그래밍을 위한 도구인 JMF (Java Media Framework) 를 사용하여 오디오와 비디오 데이터를 처리 하였다. HTML, XML 과 같이 구조적 의미를 갖는 데이터들은 구분분석기(Parser)로 구문분석 하여 해당 부분만을 DOM 인터페이스를 이용해 재구성 하였다. 구조적인 문서에서는 가상문서에서 정의된 스타일이 아닌, 원래 문서의 스타일을 적용하도록 하였다. 오디오, 비디오, 구조적 문서의 처리과정을 살펴보면 다음과 같다.

오디오, 비디오 데이터 요청 시 SS 는 해당 URL 의 정보를 이용해 전체 문서일 경우 SS 에 문서를 요청하지 않고 변환될 문서의 멀티미디어(audio, video) 태그 안의 Src 속성값으로 사용한다. 부분 문서일 경우, SS 는 가상문서에서 표현된 <ELink href="dl.cnu.ac.kr/pdoc/bach.mpeg#video(Start, Offset)" ...>중에 URI 의 Fragment 정보인 Start, Offset 의 시간 정보를 뽑아서 전체 멀티미디어 데이터 중에 해당 시간만큼 JMF(Java Media Framework)를 이용해 인코딩(encoding)한다. 인코딩된 데이터를 SS 는 임시 장소에 저장하고, UA 에 반환한다. 반환 값으로 넘겨 받은 URL 값을 문서처리기에서 변환될 문서의 멀티미디어(audio, video)태그의 Src 속성값으로 사용한다.

구조적인 문서 요청 시 SS 는 해당 URL 의 정보를 이용해 전체 문서일 경우에는 SS 에 문서를 요청하지 않고 변환될 문서의 struct 태그 안의 Src 속성값으로 사용한다. 구조적 문서의 일부분인 경우, SS 는 가상문서에 표현된 <ELink href="dl.cnu.ac.kr/pdoc/bach.html#struct(1/2/3, 1/2/5)" ...>중에 URI 의 Fragment정보인 #struct(1/2/3, 1/2/5) 의 '/'로 표현된 위치정보와 범위를 뽑아서 구문분석기(parser)를 이용해 부분문서를 생성한다. 생성된 문서를 SS 는 임시장소에 저장하고, UA 에 반환한다. 반환 값으로 넘겨 받은 URL 값을 문서처리기에서 변환될 문서의 struct 태그의 Src 속성값으로 사용한다.

4. 관련 연구

디지털 도서관 시스템에서의 멀티미디어 데이터를 지원하는 연구가 여러 기관에서 수행 되고 있다. FEDORA[Pay 98]에서의 디지털 객체 (Digital Object) 개념을 사용하여 미디어 데이터를 위한 컨테이너 두어 " disseminator"를 통해 외부와의 인터페이스를 제공한다. FEDORA 는 미디어 데이터를 처리하기위한 컨테이너 구조를 두었다는 점에서 우리의 시스템과 공통점이 있으나, 접근 방식에 있어서 링크개념을 도입하여 미디어 데이터를 처리하는 우리 시스템과는 차이가 있다. W3C에서도 웹상에서 멀티미디어를 처리하기 위한 표준으로 SMIL [SMIL98]을 정했다. SMIL 에서는 각 미디어 데이터 간에 동기화를 통하여 여러 미디어가 동시에 플레이 가능하도록 하고 있다. SMIL 에서의 <anchor> 태그를 이용한 시, 공간적인 특정부분으로의 링크 표현방식은 우리의 시스템에서의 참조링크(Rlink) 개념과 공통점이 있지만, 전체문서를 대상으로 링크를 표현하는 SMIL 과 전체문서와 부분문서를 대상으로 링크를 표현하는 우리의 시스템과는 차이가 있다. Carnegie Mellon 대학에서 개발하고 있는 Informedia Project [CMU00]는 디지털 비디오 라이브러리로서 음성인식, 영상처리, 자연어 처리 및 저작권 사용에 대한 관리 기술 등을 통합 하여 비디오 데이터의 인덱싱 및 검색에 대한 효과적인 처리 방법을 제시하고 있다. 비디오 데이터에 대하여 내용기반 검색을 하는 것이 특징이나, 가상문서의 링크와 같은 개념을 지원하지는 않는다.

5. 결론

이미 존재하는 문서들을 링크를 이용하여 새로운 형태의 문서로 만들 수 있는 가상문서는 인터넷 상에서 자유롭게 지식을 공유하고, 유용하게 이용할 수 있도록 해준다. 본 논문에서는 기존에 디지털 도서관의 문서처리기에서 지원한 텍스트, 이미지 데이터를 포함하여 멀티미디어 와 구조적 문서를 처리할 수 있도록 DTD 의 표현방식을 확장하였으며, 처리 방법을 제시하였다. 향후, 분산된 환경에서 저작권이 있는 멀티미디어 콘텐츠를 가상문서에서 사용하기 위한 연구가 필요하다.

6. 참고 문헌

[강99] 강지훈, 맹성현, 이만호, "분산 디지털 도서관 시스템에서 XML을 이용한 가상문서의 표현과 처리," 제 2 회 디지털 도서관 컨퍼런스, 서울, 1999 년 11 월.

[CMU00]Informedia Digital Video Library at Carnegie Mellon University.(http://www.informedia.cs.cmu.edu), 2000

[DOM98] W3C, Document Object Model (DOM) Level 1, Recommendation, Oct. 1998. (http://www.w3.org/TR/REC-DOM-Level-1).

[Mya99] S. H. Myaeng, M.-H. Lee, and J-H. Kang, "Virtual Documents: a New Architecture for Knowledge Management in Digital Libraries, " Proc. Asian Digital Libraries Conf., Taipei, Taiwan, Nov. 1999.

[Pay98] S. Payette, and C. Lagoze, "Flexible and Extensible Digital Object and Repository Architecture," Second European Con. on Research and Advanced Technology for Digital Libraries, Heraklion, Crete, Greece, Sept. 1998,. (http://www.cs.cornell.edu/payette/papers/ecdl98/fedora.html)

[Mya00]S. H. Myaeng, M.-H. Lee, J.-H. Kang, E.-I. Cho, Y.-B. Lee, D.-S. Lim, J.-M. Lim, H.-J. Oh, and J.-S. Yang, "A Digital Library System for Easy Creation/Manipulation of New Documents over Existing Resources", RIAO 2000, Paris, France, April 2000.

[SMIL98]Synchronized Multimedia Integration Language (SMIL) 1.0, Recommendation, June. 1988(http://www.w3.org/TR/REC-smil).

[XML98] W3C, Extensible Markup Language (XML) 1.0, Recommendation, Feb. 1998. (http://www.w3.org/TR/REC-xml).

[XSL99] W3C, Extensible Stylesheet Language (XSL) Specification, Working Draft, Apr. 1999. (http://www.w3.org./TR/WD-xsl)