

XML 기반 가상문서의 브라우저를 위한 문서 처리

*양중식, **임동수, **강지훈
*고려정보테크(주), **충남대학교 컴퓨터과학과
e-mail : jsyang@koreait.co.kr, {lastdkht,jhkang}@cs.chungnam.ac.kr

Document Processing for Browsing XML-Based Virtual Documents

*Jung-Shik Yang, **Dong-Soo Lim, **Ji-Hoon Kang
*Korea IT Inc. **Dept. of Computer Science, Chungnam National University

요 약

XML을 기반으로 하는 가상문서는 인터넷 상의 정보 공유를 가능하게 하여 새로운 지식을 생성할 수 있도록 한다. 본 논문에서는 가상문서를 지원하는 디지털도서관 시스템에서 XML로 표현된 가상문서를 효과적으로 처리하도록 하는 문서처리를 설계 구현하였다. 클라이언트의 요청에 따라 문서 처리 시스템은 가상문서를 클라이언트 브라우저에서 볼 수 있는 형태로 변환하여 전달한다. 그러므로 가장 중요한 기능은 내포링크, 참조링크, 총칭링크 등 다양한 의미의 링크들을 포함하는 가상문서를, 변환을 통하여 클라이언트에게 정확하게 전달하는 것이다. 추가적인 기능으로 우리의 디지털도서관 시스템의 다른 구성 요소인 저장 서버, 검색 서버, 링크 서버, 이름 서비스 등을 지원한다.

1. 서론

인터넷의 급격한 발달과 더불어 문서의 배포와 공유의 속도는 날로 더 가속화 되어가고 있으며, 그 범위 역시 개인 가정에만 이르고 있다. 이러한 분산 환경에서 새로운 문서를 만들 경우, 인터넷 상의 여러 곳에서 문서들의 일부 또는 전부를 가지고 와서 사용하는 경우가 종종 있다. 인터넷은 하나의 커다란 지식 창고라고 할 수 있으며 문서를 손쉽게 공유할 수 있는 장점을 가지고 있다. 그러므로 디지털 도서관은 디지털 문서를 검색하고 전문을 제공하는 것과 같은 본연의 기능 이상의 것을 수행하기를 요구 받는다. 즉, 문서로 되어 있는 지식을 쉽게 공유하고 생성할 수 있는 방법을 제공할 필요가 있다. 이러한 지식 생성 기능을 분산 디지털 도서관 시스템에서 효율적으로 지원하기 위하여 제안된 새로운 문서 개념이 가상문서 (virtual document)이다 [Myaeng 99].

인터넷 상의 문서를 일부 또는 전부를 링크로 연결함으로써 새로운 문서를 생성할 수 있는 가상문서는 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 [Myaeng 99].

이러한 가상문서를 디지털도서관 시스템이 지원하기 위해서는 정형화된 가상문서 양식이 필요하며, 표현력, 처리 효율성, 인터넷 접근성 등을 고려할 때 XML [XML 98]이 적합하다고 판단하였으며, 이에 따라 가상문서의 양식과 관련 스타일시트 양식이 우리의 선행 연구를 통하여 XML DTD로 정의되었다 [Myaeng 99, 강지훈 99].

* 이 연구는 충남대학교 소프트웨어연구센터의 재정지원을 받았다.

본 논문에서는, 이렇게 가상문서를 지원하는 디지털도서관 시스템에서 XML로 표현된 가상문서를 효과적으로 처리하도록 하는 문서처리 시스템을 설계 구현하였다. 클라이언트의 요청에 따라 문서 처리 시스템은 가상문서를 클라이언트 브라우저에서 볼 수 있는 형태로 변환하여 전달한다. 그러므로 가장 중요한 기능은 내포링크(embedding link), 참조링크(referential link), 총칭링크(generic link) 등 다양한 의미의 링크들을 포함하는 가상문서를, 변환을 통하여 클라이언트에게 정확하게 전달하는 것이다. 부가적인 기능으로 우리의 디지털도서관 시스템의 다른 구성 요소인 저장 서버, 검색 서버, 링크 서버, 이름 서비스 등을 지원한다.

본 논문의 구성은 다음과 같다. 2 절에서는 가상문서에 대하여 설명하며, 3 절에서는 가상문서의 처리 시스템의 설계와 구현에 대하여 설명한다. 마지막으로 4 절에서 결론을 맺는다.

2. 가상문서

2.1 다양한 링크를 지원하는 가상문서

가상문서는 링크로만 구성된 문서이다. 기본적으로 가상문서의 구성을 나타내는 내포링크와, 단순한 참조를 나타내는 참조링크로 이루어져 있다. 한편 방향성, 목적지의 개수 등에 따라 양방향 링크, 다중 링크, 그리고 여러 개의 물리적인 링크를 하나의 개념적인 링크로 표시할 수 있는 총칭링크 등이 있다. 그러므로 가상문서의 처리에 있어서 핵심적인 내용은 링크의 의미에 따라 정확히 처리하는 것이다.

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 불러와서 하나로 취합이 되어 온전한 문서 형태로 클라이언트에게 보여야 한다. 그러므로 내포링크들은 문서 안에서 순서가 중요하다.

참조링크는 링크의 다양한 방향을 지원한다. 기존의 HTML 문서에서 사용하던 단방향 링크는 물론, 양방향링크(bi-directional link), 목적지가 여러 곳인 다중링크(multi-directional)를 지원한다.

한편 가상문서는 참조 링크의 출발지가 여러 개로 구성된 링크를 지원한다. 예를 들어 가상문서에 내포되는 문서에 나타나는 "W3C"라는 모든 단어는 W3C의 홈페이지를 가리키도록 하는 링크이다. 이러한 링크를 총칭 링크(generic link)라고 한다. 그러므로 출발지는 특정 위치를 나타내는 것이 아니라, 출발지를 의미하는 표현식이 된다.

가상문서에서는 HTML에서처럼 링크를 출발지에 함께 두는 in-line 링크가 아니다. 링크의 출발지나 목적지와는 별도의 곳인 가상문서 안에, 즉 문서의 실체와는 떨어진 곳에 링크를 두는 out-of-line 링크를 사용한다. 이것은 CD-ROM과 같은 읽기 전용 속성을 가진 문서를 물리적으로 변경하지 않고도 새로운 링크를 그 문서에 부여할 수 있게 하여 준다.

<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] 가상문서의 DTD.

2.2 XML DTD로 정의된 가상문서 양식

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

가상문서는 허브(hub)와 스타일시트(stylesheet)로 구성된다. 허브는 문서의 구조와 내용, 그리고 스타일시트는 가상문서 전체 및 그 구성요소를 위한 스타일 정보를 가지고 있다. 허브는, 내포링크 순서(a sequence of embedding links), 참조링크 집합(a set of referential links), 그리고 메타데이터(metadata)의 세 개로 구성된다.

VdocHub는 가상문서에서 가장 상위 요소(Element)로 ELinkSeq, RLinkSet, Metadata의 순서를 갖는 세 개의 요소를 가지고 있다. ELinkSeq는 가상문서에서 문서가 삽입되는 단위들인 ELink가 한 번 이상 나타난다. ELink 요소는 디지털 도서관 시스템에서 검색 및 링크서버에서 문서를 관리와 처리하는 단위가 된다.

ELink 요소는 가상문서, 텍스트, 비디오, 오디오, 이

미지 파일 등의 다양한 문서가 포함될 수 있으며, 이러한 문서의 부분적인 내포가 가능하다. ELink 속성값 href는 문서가 내포될 목적지를 나타내며 문서의 전체 또는 일부를 가리킬 수 있으며 출발지는 ELink 자신이 된다. 가상 문서에서 사용되는 URI표기 방법과 부분 표기 방법을 보면 다음과 같다.

문서가 텍스트일 때는 아래와 같이 # 다음에 text라는 한정어를 두고 다음 Start와 Offset으로 문서의 부분을 표기했다. Start는 부분 문서가 시작되는 byte이고, Offset는 문서 시작 위치에서부터 포함될 문서의 끝 부분을 나타내며 공백 문자를 포함한다.

<ELink href="dl.cnu.ac.kr/pdoc/bach.txt#text(Start, Offset)" ...

문서가 그림일 경우에는 # 과 한정어 rectangle다음에 [그림 2]과 같이 각 좌표 값 x1, y1, x2, y2 를 표기한다.

<ELink href="dl.cnu.ac.kr/pdoc/bach.gif#rectangle(x1, y1, x2, y2)" ...

XML또는 가상문서 같은 구조적인 문서의 부분이 내포될 경우에는 문서의 구조적 경로로 문서의 요소를 가리킨 후 해당 요소 내용의 Start, Offset으로 표기하였고 해당 요소만 표시될 경우에는 Start, Offset을 생략할 수 있다.

<ELink href = "dl.cnu.ac.kr/pdoc/bach.xmlt#1/2/2(Start, Offset)" ...

ELink 속성 중 role과 title는 링크의 의미적인 정보를 표시하며, category 속성은 내포되는 문서가 어떤 분류에 속하는지 표시해 준다. owner와 date 속성은 링크를 만든 사람과 날짜를 위한 목적으로 사용된다. actuatedefault 속성은 가상문서가 사용자에게 보여질 때 문서가 사용자 요청에 의해 내포(user), 또는 자동으로 내포될 것인지(auto)에 관한 선택으로 사용자 요청에 의해 문서가 내포될 경우에는 문서가 위치할 곳에 title의 내용이 위치하게 되고 사용자는 링크로 표시된 title을 클릭하여 문서를 얻을 수 있다. 이 속성은 내포하려는 문서의 크기가 클 경우 문서를 선택적으로 볼 수 있게 해 주는 기능을 가지고 있다. autoDelete 속성은 내포되는 문서가 삭제되는 등의 이유로 더 이상 유효하지 않을 경우 디지털 도서관 시스템에서 자동으로 링크를 삭제하는 것을 허락할 것인가에 대한 것을 표시하며 기본적으로 삭제를 허락하지 않는다.

RLinkSet은 0 개 이상의 RLink의 요소로 구성되어 있으며, RLink는 다시 하나의 Source 요소와 1 개 이상의 Destination 요소로 이루어져 있다. RLink 요소는 내포되는 문서에 링크를 표시하지 않고, 내포되는 문서와는 별도로 링크를 정의할 수 있는 out-of-line 링크를 사용하여 링크를 표시할 수 없는 일반 텍스트 문서나 CD-ROM 같은 읽기 전용 문서에 링크를 표시할 수 있다. 하나의 링크 대상에 대하여 여러 개의 목적을 가리킬 수 있는 다중링크(Multidirection Link)도 표현 가능하다. RLink는 showdefault와 actuatedefault타입을 가지고 있는데, showdefault는 사용자가 링크를 선택했을 때 목적지의 문서를 어떻게 보여줄 것인지를 표시한다. new는 새 창을 띄워서 문서를 보여주며, parsed는 현재 문서 사이

에 내포되고, replace는 현재창의 내용을 지우고 대상 문서를 보여주는 기능을 가진다. actuatedefault는 ELink 요소에서 설명한 내용과 같은 의미를 가진다.

Source 요소는 내포문서의 링크 대상을 나타내는 것으로 링크 방법에 따라 두 가지로 나뉘어진다. 첫 번째로는 가상문서에 나타나는 특정 어구에 대하여 링크를 표시하는 총칭링크(Generic Link)를 가지고 있다. 총칭링크는 Is_generic 속성이 반드시 YES가 되어야 하며, 삽입하려는 특정 어구는 href 속성값으로 표시한다. 예를 들어 “컴퓨터” 라는 단어에 총칭링크를 나타내고자 하면 아래와 같이 나타낼 수 있다.

<Source Is_generic = “YES” href = “컴퓨터” ...

Destination 요소는 내포문서에 추가한 링크의 목적지를 나타내는 것으로 ELink의 속성들과 같은 의미를 가지는데, 다중링크를 나타내거나 RLink actuatedefault 요소가 user일 때 title정보로 링크의 목적지를 나타내고, 사용자가 그것을 선택했을 때 목적지의 문서를 볼 수 있게 된다.

VdocHub의 마지막 요소인 Metadata는 DublinCore [Baker 98]에 나오는 15 개의 각 항목에 대해 요소로 선언했으며, 가상문서에 대한 부가적인 정보를 기술하는 역할을 한다.

[그림 2]는 가상문서의 스타일을 위한 DTD를 나타내며 VdocStyle은 하나이상의 ELinkStyle로 구성된다. ELinkStyle의 개수는 보통 ELinkSeq에 있는 ELink 수와 동일하다. ELinkStyle은 내포되는 문서에 대한 스타일 정보를 제공한다. ELinkStyle은 내포될 문서의 위치를 나타내는 align과, 글씨의 font, 내포될 문서의 색 정보를 나타낸다. align은 내포되는 문서의 위치인 왼쪽, 중간, 오른쪽 중 하나를 선택할 수 있으며, font 요소는 글씨체 이름, 크기, 볼드, 이탤릭의 네 가지 속성을 줄 수 있다. 글씨 색에 대한 속성에는 #0000FF(파란색)와 같이 색상 값을 사용한다.

<ELEMENT	VDocStyle	(ELinkStyle+)	>
<ELEMENT	ELinkStyle	(align? font? color?)	>
<ELEMENT	align	EMPTY	>
<ATTLIST	align		
	value	(left center right)	"left" >
<ELEMENT	font	EMPTY	>
<ATTLIST	font		
	name	DATA	#IMPLIED
	size	CDATA	#IMPLIED
	bold	(true false)	"false"
	italic	(true false)	"false"
<ELEMENT	color	EMPTY	>
<ATTLIST	color		
	value	CDATA	#REQUIRED >

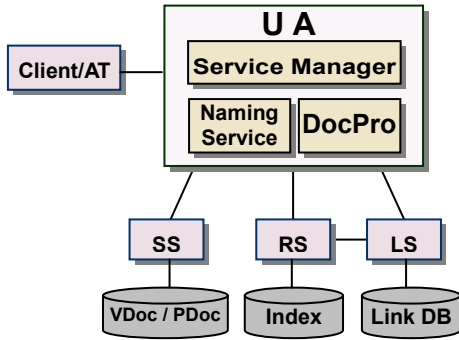
[그림 2] 가상문서 스타일시트를 위한 DTD.

3. 가상문서 처리

3.1 문서처리 모듈의 구조

우리의 디지털 도서관 시스템[Myaeng 99]에서 문서처리 모듈(Document Processing Module)인 DocPro

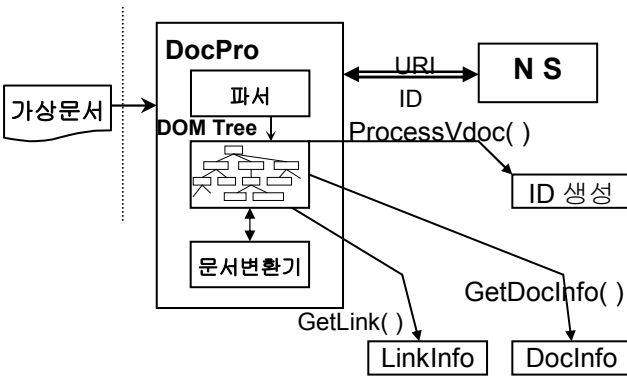
는 UA(User Agent)의 한 부분으로 가상문서의 처리를 담당한다. 문서처리는 가상문서의 고유 ID(Identifier) 생성과, 링크서비스에 필요한 링크정보를 반환하며, 문서의 검색을 위한 가상문서 ID와 문서에 정의된 메타데이터, 내포 문서들과 그 문서에 대한 ID들을 반환해준다. 그리고 DocPro의 주기능인 문서변환의 기능을 가지고 있다. [그림 3]은 디지털 도서관시스템에서 가상문서 처리를 위한 UA의 구조를 나타내고 있다.



[그림 3] 가상문서 처리를 위한 UA의 구조

3.2 가상문서의 ID, 링크정보, 문서정보 생성

가상문서의 ID생성, 링크정보 및 문서정보생성은 저작도구를 통하여 문서가 생성되어 문서가 저장될 때 이루어지는 작업이다. [그림 4]는 DocPro의 기능 중 문서변환 기능을 제외한 ID생성, LinkInfo, DocInfo생성 과정을 나타낸 것이다.



[그림 4] 가상문서 저장에 따른 DocPro의 동작.

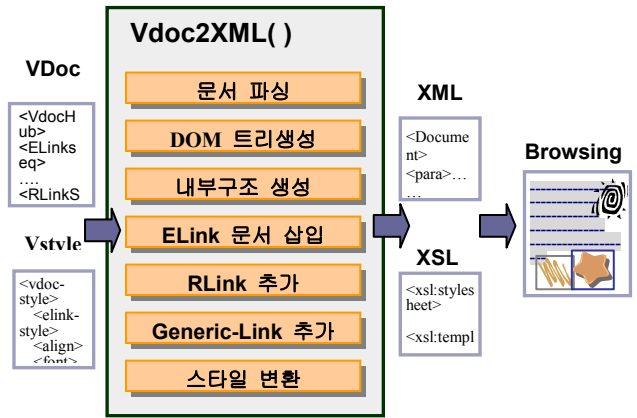
가상문서의 저장시 처음으로 이루어지는 과정인 가상문서 ID생성 알아보면 다음과 같다. DocPro는 SM(Service Manager)으로부터 가상문서를 전달 받아 XML 파서를 이용해 DOM[DOM 98] 트리를 얻어낸다. DOM 트리를 탐색해 가상문서에 내포되는 문서에 대한 URI를 얻어내어 NS(Naming Service)를 통해 ID를 생성한다. BlockID는 가상문서를 위한 디지털 도서관 시스템에서 효율적인 검색과 링크 관리를 위한 문서 단위로, 가

상문서에 내포되는 각각의 문서가 BlockID가 될 수 있으며, 가상문서가 가상문서를 내포하는 경우에는 가상문서 전체가 BlockID가 된다.

LS(Link Server)를 위한 링크정보 생성은 가상문서의 BlockID가 성공적으로 생성된 후 DocPro내의 GetLink 매소드를 통해 이루어진다. GetLink 매소드는 DOM 트리를 이용해 링크 정보를 생성한 후 SM을 통해 LS에 저장 된다. RS(Retrieval Server)를 위한 문서정보 생성도 GetLink와 마찬가지로 가상문서의 ID가 성공적으로 생성된 후 SM이 DocPro내의 GetDocInfo 매소드를 호출에 의해 이루어진다. GetDocInfo 매소드는 가상문서 내에 기술되어있는 메타데이터 정보를 DOM 통해 얻어내고 가상문서에 내포되는 문서와 문서에 대한 BlockID를 생성하여 DocInfo에 저장한다. 가상문서에 내포되는 문서는 검색서버에서 색인을 위한 것으로 가상문서의 RLinkSet에 정의된 링크정보가 내포문서에 추가 되지 않은 문서이다.

3.3 가상문서의 변환

가상문서를 처리하는 문서처리기에서 가장 중요한 역할을 하는 문서 변환기는 링크요소로만 되어있는 가상문서를 사용자가 가상문서를 작성한 의도대로 볼 수 있도록 한다. DocPro 내에 있는 문서 변환기의 내부구조는 [그림 5]와 같다.



[그림 5] XML 문서 변환기 구조.

가상 문서가 입력되어 변환되기까지의 과정을 알아보면, 문서처리기에서 가상문서를 받아와 Vdoc2XML에 건네준다. Vdoc2XML은 가상문서 내에 정의한 문서를 파싱하여 문서정보를 추출한다. Vdoc2XML에서는 효율적인 문서 변환을 위하여 내부 자료구조를 두었는데, 내부 자료구조에 저장될 정보에 대한 설명은 다음과 같다. ELinkInfo는 가상문서에서 ELinkSeq의 자식 요소인 ELink에 대한 정보로 내포될 문서의 URI, title, actuatedefault, category등의 속성값이 저장되며, 내포되는 문서의 스타일 정보가 함께 저장된다.

가상문서의 스타일은 가상문서가 내포되는 문서의 단위로 정의가 되며 내포문서의 정렬 값, 글자의 크기, 색, 폰트, 이탤릭체 등의 스타일 정보가 ELinkInfo에 ELinkStyle의 타입으로 저장되는데 ELinkInfo는 내포되는 문서의 수만큼 생성된다. RLinkInfo는 링크 정보를 추가할 대상이 되는 내포문서 순서 값(Order)과 RLink 지식 요소인 Source와 Destination의 속성값을 저장한다. RLinkInfo에 저장되는 Fragment 정보는 내포되는 문서의 형식에 따라 다르기 때문에 링크를 추가할 문서의 타입에 따라 Fragment 객체를 상속 받아 해당하는 Fragment 정보를 저장한다. 본 논문에서 지원하는 내포되는 문서는 이미지 문서, 텍스트문서, 그리고 가상문서와 XML 같은 구조적인 문서로 RLinkInfo에 들어가는 Fragment 타입은 세 가지이다. GLinkInfo는 가상문서에서 얻어낸 File Generic Link 정보와 LS에서 얻어낸 Site Generic Link 정보를 저장하는데, 문서에 링크를 추가할 특정 어구(Term)와 그에 대한 목적지 문서의 제목과 URI값이 저장된다.

문서 변환기는 가상문서를 Well-formed XML 문서로 변환한다. 변환을 위한 Vdoc2XML의 내부구조가 완성되면 문서 변환의 첫번째 과정으로 ELinkInfo에 저장되어 있는 URI를 얻어와 SS에 문서를 요청한다. 문서를 요청할 때는 항상 자신의 저장서버에 문서가 있는지 확인 후 그 결과에 따라 문서 요청하는 작업이 선행되어야 한다.

내포될 문서가 전체 이미지 경우에는 SS에 문서를 요청하지 않고 ELink에 정의된 URL을 변환될 문서의 Image태그 안의 Src 속성값으로 사용하고, 이미지의 일부분 인 경우에는 SS에 문서를 요청하여 반환 값으로 넘겨받은 URL값을 Image 태그의 Src속성값으로 사용한다. 내포될 문서가 텍스트 문서의 경우에는 내포될 문서가 전체 또는 부분에 상관없이 SS로부터 넘겨받는 문서의 스트링 값을 추가한다. 문서의 형태가 가상문서의 경우에는 재귀적으로 Vdoc2XML을 호출하여 변환된 XML 문서를 추가한다. 위와 같이 가상문서에 내포되는 문서들의 추가작업은 가상문서에 정의되어있는 ELink의 순서대로 진행된다. 링크를 삽입한 후, 변환될 Para태그 안에 가상문서에서의 ELink 순서 값을 ID값으로 부여하여 스타일 적용시 이용하게 된다.

가상문서에 내포되는 문서 삽입과 RLink작업은 내포되는 문서단위로 작업이 진행된다. 문서 삽입 작업이 끝나면 RLinkInfo에 저장된 문서의 Fragment정보를 얻어와 링크 삽입위치에 Src 태그를 삽입하고 Destination 요소의 href 속성 값이 저장 되어있는 RLinkInfo의 Dst.값을 얻어와 Anchor 태그의 요소인 href속성값으로 취한 후 태그를 삽입한다. Destination이 여러 개 있을 경우에는 Anchor 태그 뒤에 이어서 삽입할 수 있다. Anchor요소를 삽입한 후 아래 예제와 같이 링크의 대상이 되는 곳에 Src 태그를 붙이고 링크의 목적지에 Dst 태그를 삽입한다. Link요소는 ID 속성을 두어 브라우저링을 위한 XSL[XSL 99] 스타일시트에 적용할 수 있도록 하였다.

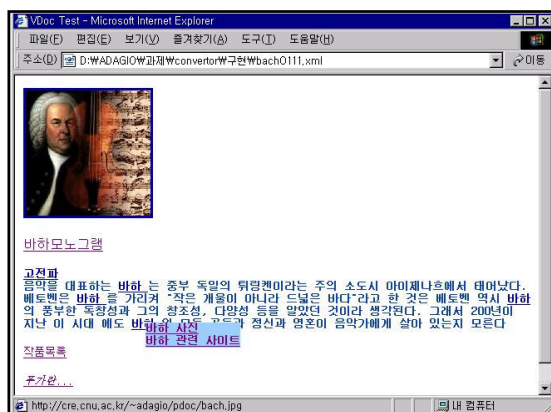
..... 간결하고 요령있는 <Link id="e1r1"><Src>푸가</Src> <Dst>
<Anchor href="http://dl.cnu.ac.kr/fuga.txt" title="푸가용

어" showdefault="replace" />
<Anchor href="http://dl.cnu.ac.kr/worklist.txt" title="푸가작품" showdefault="replace" />
</Dst>
</Link>의 서법을 배웠다.

내포되는 문서에 RLink를 삽입한 후 특정 어구(Term)에 링크를 추가하는 Generic Link를 추가하기 위해 GLinkInfo에 저장되어있는 Term을 얻어와 RLink가 추가된 문서에서 Term과 일치하는 곳을 찾아낸다. Generic Link외의 모든 정보가 추가된 문서에는 태그 내에 Title정보나 이미지의 속성 값으로 링크의 목적지가 되는 대표 제목을 붙여있기 때문에 이와 같은 값에 Generic Link정보가 삽입되지 않도록 확인하는 작업을 거친 후 문서의 RLink를 추가했을 때와 같은 방법으로 문서에 링크를 추가한다. 아래 예제는 내포된 문서 안에서 Term이 '바하' 인 경우 링크가 추가된 예를 보여주고 있다.

..... 삼촌도 교회 오르가니스트였다. <Link id="t1g2"><Src>바하</Src> <Dst>
<Anchor href="http://dl.cnu.ac.kr/bach.jpg" title="바하 사진" showdefault="replace" />
</Dst></Link>는 이러 한 음악적인
현대의 서양 클래식 음악이 모두 한꺼번에 사라진다 해도 <Link id="t1g3"> <Src>바하</Src> <Dst>
<Anchor href="http://dl.cnu.ac.kr/bach.jpg" title="바하 사진" showdefault="replace" />
</Dst></Link>의 [평균율 클라비어 곡집]만 남아 있다면 현재의 음악을

가상문서를 사용자가 브라우저에서 볼 수 있도록 하기 위해서는 XSL 스타일시트도 변환된 문서에 맞게 변환 해주어야 한다. 현재 사용되는 브라우저는 가상문서 개념에서 사용되는 다중링크나, Embedding Link등의 기능을 지원하고 있지 않기 때문에 가상문서 변환기에서는 XSL과 자바스크립트를 이용해 다중링크와 Embedding Link를 표현하고 있다.

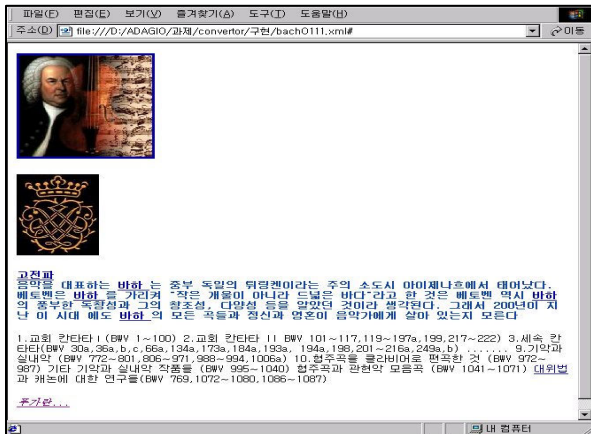


[그림 6] 변환된 문서를 IE 5.0에서 브라우저링한 화면

변환된 가상문서는 마이크로소프트의 인터넷 익스플로러 5.0 이상 버전에서 브라우저링이 가능하고 가상문서

에서 정의한 기능들을 사용할 수 있다. [그림 6]은 변환된 가상문서를 브라우저한 모습이며 다중 링크를 클릭한 모습이다.

[그림 7]은 ELink 요소의 actuatedefault속성 값에 따라 변환된 문서가 화면에 브라우징 되는 모습을 보여주고 있으며, [그림 6]에서 두 번째 내포문서와 네 번째 내포문서를 보기 위하여 선택했을 때 문서가 브라우징 된 모습이다.



[그림 7] 내포문서가 보여지는 화면

[그림 6]과 [그림 7]에서 보듯이, actuatedefault 속성 값에 따라 문서를 보여주는 모습이 다르게 결정된다. 그 값이 'user' 인 경우에는 ELink 요소의 title 속성 값만을 보여주며 화면의 링크를 클릭해야 문서를 볼 수 있다. 문서가 'auto' 일 때는 자동으로 문서를 보여준다.

4. 관련 연구

가상문서 개념과 관련이 있는 연구가 첨단 디지털 도서관 기술 개발을 주도하는 연구 기관들에서 최근에 수행되어 오고 있다. 다가문서 (multivalent document) 모델 [Wilensky 98] 과 FEDORA [Payette 98] 에서의 디지털 객체 (Digital Object) 개념은 디지털 도서관을 위한 문서 구조를 정의한다는 점에서 우리의 시스템과 공통점이 있다. 전자의 경우, 문서는 다양한 형태의 층으로 이루어지며, 각 층은 "behavior"라 부르는 프로그램의 지원을 받는다. 후자의 경우에는, 디지털 객체는 사용자 측면에서는 내부 구조가 감추어져 캡슐화 되어 있으며 "disseminator"라 부르는 서비스 인터페이스의 집합을 통하여 사용자가 디지털 객체에 개념적으로 쉽게 접근할 수 있도록 하고 있다.

우리의 가상문서 모델은 "behavior"나 "disseminator"라는 처리 개념이 없는 좀 더 단순한 형태를 취하고 있다. 반면, 다양한 링크를 지원함으로써, 기존의 문서를 재사용 하여 새로운 문서를 쉽게 만들 수 있다. 한편, 기존의 두 접근 방식은 자료를 틀 안에 넣고 구조화하는 데에 주안점을 둔 데 비하여, 우리의 방식은 자료를 연결하고 확장하는 데에 초점을 맞추었다.

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 XML에 기반한 가상문서의 문서 타입 정의와 문서 처리 및 브라우징을 위한 문서처리와 변환에 대한 설계와 구현 방법에 대하여 알아보았다. 구현된 문서처리 및 변환 시스템은 충남대학교에서 개발한 디지털도서관 시스템인 MIRAGI-III의 한 부분으로서 문서 처리 및 변환을 담당하고 있다. 분산 처리를 위하여 전체 디지털도서관 시스템이 CORBA 환경으로 구현되어 있으며, 프로그램의 이식성을 위하여 Java언어를 사용하였다.

현재는 가상문서가 텍스트와 이미지만을 지원하도록 되어 있으나, 향후 동영상을 포함하는 다양한 멀티미디어를 지원하도록 강화할 예정이다. 그에 따라 부분 문서를 표기하는 표기법도 확장될 것이다.

참고 문헌

[강지훈 99] 강지훈, 맹성현, 이만호, "분산 디지털 도서관 시스템에서 XML을 이용한 가상문서의 표현과 처리," 제 2 회 디지털 도서관 컨퍼런스, 서울, 1999년 11 월.

[Baker 98] T. Baker, "Language for Dublin Core," D-lib Magazine, Dec. 1998.

[DOM 98] W3C, Document Object Model (DOM) Level 1, Recommendation, Oct. 1998. (<http://www.w3.org/TR/REC-DOM-Level-1>).

[Myaeng 99] 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.

[Payette 98] 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, September 21-23, 1998. (<http://www.cs.cornell.edu/payette/papers/ecdl98/fedora.html>)

[Wilensky 98] Robert Wilensky and Thomas A. Phelps, "Multivalent Documents: A New Model for Digital Documents," Technical Report, CSD-98-999 (1998)

[XML 98] W3C, Extensible Markup Language (XML) 1.0, Recommendation, Feb. 1998. (<http://www.w3.org/TR/REC-xml>).

[XSL 99] W3C, Extensible Stylesheet Language (XSL) Specification, Working Draft, Apr. 1999. (<http://www.w3.org/TR/WD-xsl>)