Virtual Documents: a New Architecture for Knowledge Management in Digital Libraries[1]

Sung Hyon Myaeng, Mann-Ho Lee, & Ji-Hoon Kang
Department of Computer Science
Chungnam National University
Taejon, Korea
{shmyaeng, mhlee, jhkang}@cs.chungnam.ac.kr

Abstract

Digital libraries are often viewed as a collection of documents from which users can retrieve documents for their needs. As an extension to this conventional notion of digital libraries, we introduce a new document architecture where links play a key role in helping users create knowledge. It allows for easy creation of a new composite document, called a virtual document, whose parts can be geographically dispersed. This paper describes the concept of virtual documents and a digital library system architecture geared toward applications where an essential element is teacher-student interactions through an exchange of knowledge in the form of multimedia documents.

1. Introduction

Digital libraries are often viewed as a collection or a repository of documents from which users can retrieve documents for their needs. Unlike conventional information retrieval systems, however, documents that form a collection can be stored in distant locations. As such, previous research has focused on the problems associated with digitization of materials, storage of a large amount of documents, and retrieval from distributed databases although such issues as right management and preservation have surfaced as unique to digital libraries. Very little attention has been paid to the issue of making digital libraries as a workspace where new knowledge can be created.

As an extension to the conventional notion of digital libraries where documents remain as independent entities even with hypertext links, we introduce a new document architecture with which digital libraries can be viewed as a dynamic knowledge space. By *dynamic*, we mean that users can easily create new documents from existing ones with various links. By *knowledge*, we mean inter-connection of independent documents or their parts using links. Unlike the traditional retrieval situation where even documents connected by hypertext links are all treated as independent ones, we propose

[1] This research has been supported by Software Research Center, Chungnam National University.

a method of retrieving knowledge, i.e., connected documents as a unit. The term space indicates that we can apply a metric to measure the distance between two pieces of knowledge (i.e. between two groups of interconnected documents). The proposed document architecture allows for easy creation of a new composite document, called a *virtual document,* whose parts are connected with different types of links but can be geographically dispersed in several locations.

Our attempt is unique in that we focus on easy creation of new documents based on existing ones with different types of links, currently embedding links and referential links. An *embedding link* is used to insert a pointer to an existing document or its part to the virtual document being created without having to copy it. When a virtual document is *instantiated* to be displayed on a screen, for example, the parts connected by embedding links are actually copied. A referential link, on the other hand, acts like the usual hypertext links found in web documents; a document pointed to by a referential link is shown by user activation. However, referential links can be made activated automatically like embedding links. Another interesting aspect of our approach is to allow great flexibility for links. The source or the destination of a link can have more than one object that can be either a whole document or a part.

The "multivalent document model" [Phe96] and "Digital Objects" in the FEDORA framework [Phe98] share some commonality with our work in that they define their own document architectures for digital libraries. In the former approach, documents are viewed as layers of content, possibly of different types, and supported by dynamically loaded program objects, called "behaviors". In the latter, a Digital Object is a combination of an abstraction for a content container and a kernel that encapsulates content as opaque byte stream packages and an interface layer that gives contextual meaning to the content. Compared to the multivalent document model, the FEDORA approach segregates the structure, content-type, interfaces, and mechanism that execute content-type behavior, with right management as the main goal.

In contrast, our virtual document model has a simpler structure emphasizing not on the notion of behaviors or program objects, but on the use of diverse types of links, for easy creation of new documents by conveniently reusing existing documents. From a different perspective, the two previous approaches focus on packaging and structuring data whereas our approach helps creating connections and expansions.

We are currently in the process of implementing a digital library prototype supporting the virtual document architecture. The document model is realized with XML since it is becoming a de facto standard for Web documents. Among the several components that comprise a digital library system, the link server controls a link

database. All the links in individual virtual documents plus some global links[2] are stored in the link database and managed by the link server. Other components are: a retrieval server that accesses an index and retrieves documents for a given query; a storage server that manages registration and storage of documents, both virtual and physical; and a user agent that modulates all the servers to for user operations initiated by the user client.

We envision that the proposed document architecture and the system architecture described in Section 3 can be best used in the education area. With the growing interest in cyber education in which teachers and students exchange their own materials over the Web, the document model can be of practical use. It allows the teachers and students to easily utilize the materials in public digital libraries and/or in private digital libraries, for the purpose of creating new materials such as lecture notes, term papers, and class presentation materials.

2. The virtual document

The main motivation for the virtual document concept is to provide an easy way for users to reuse online digital documents or their parts to create a new document or a new view. As such, we need a coherent mechanism for defining an aggregation of possibly dispersed documents or their parts in a distributed environment. The basic mechanism in the virtual document architecture is the use of links that allow diverse ways of constructing a new document that consists mainly of links or pointers to other documents or their parts.

New functionality can be realized in digital library settings with the virtual document architecture. The key benefits that can be obtained are:

- A new composite document can be created without having to copy all the components, some of which may be multi-media materials that require a significant amount of storage or transmission time when they are located in a remote place.
- It is easy to associate meta-information on the aggregated documents. In particular, annotations created by readers can be systematically incorporated with relative ease.
- Versions or representations of documents can be handled in a principled way.
- Links can be easily created on read-only documents such as those in a CD-ROM, without having to make an extra copy for manipulation.
- With a link DB and a link server, relationships not specific to a single document but

[2] A global link is not specific to a particular virtual document and hence cannot be stored in a single document. The concept is explained in the latter part of Section 2.

applicable to documents across network can be expressed and processed efficiently.

2.1. Links

A link is defined to connect one or more source anchors to one or more destination documents or their parts. There are currently two kinds of semantics on links: referential and embedding. *Referential links* are used to indicate that the destination is a reference and can be reached usually by the user's clicking on the anchor. *Embedding links* are used to indicate that the destination is to be embedded in the document containing the source when the virtual document is instantiated.

Regardless of link semantics, links can be classified along three dimensions. Depending on the cardinality of the source and the destination, a link is either:

- one-to-one when there is only one document or part on the source and destination,
- one-to-many when there is more than one documents or parts on the destination,
- many-to-one when there are more than one anchors on the source and only one document on the destination, or
- many-to-many when there are more than one anchors on the source and more than one documents on the destination.

A one-to-many link can be used with a referential or an embedding link, but with an embedding link, there should be a way of specifying how the "many" destination parts will be embedded (or displayed when they are embedded).

Depending on the way the source is defined, a link is either: a *specific link* when there is one specific anchor, or a *generic link* when there are more than one anchor, defined by a formula. A generic link can be used in conjunction with either a referential link or an embedding link. On the other hand, depending on the way the destination is defined regardless of the cardinality, a link is either *total* when the destination is a whole document, or *partial* when the destination is a part document. Both partial and total links can be used in conjunction with a referential link or with a embedding link.

2.2. Definition of Virtual Documents

A virtual document consists of a hub that defines its structure and a style sheet that defines how it is to be shown when instantiated. A hub is defined over a set of documents, *D*, in a distributed environment. A document is either a virtual document or a physical document[3] consisting of one or more of the media types: text, graphics,

[3] A physical document may contain both referential links and embedding links as in an HTML document with an inline image, but not a hub.

image, audio, or video. More formally, a hub is a triple, <*E, R, M*>

- *E* is a sequence of one or more embedding links to documents in *D* or their parts. Each link specification must include the link attributes and the destination. The order in which the links appear is significant in that it determines the default layout of the virtual document. An embedding link may specify whether the destination is local (those owned by the author or those in the personal digital library) or non-local. For efficiency reasons, automatic embedding can be restricted to local destinations only.
- *R* is a set of referential links to documents in *D* or their parts. Each referential link specification must include the link attributes, the source, and the destination. Since the primary use of this set is to specify additional links on read-only documents, both the source and destination must be defined on physical documents. A referential link can be destined to a specific document whose role is to collect user annotations about the virtual document.
- *M* is for the meta-data section of the entire virtual document, which currently contains Dublin Core and plus a list of index terms but can be extended as needed in the future.

A *style sheet* defines the way the component documents are laid out and the whole virtual document must be displayed. Documents or their parts used as a destination of an embedding link are allowed to be superimposed as layers when their layout is specified. This is useful when other *D* parts need to be overlaid on top of existing parts. There are at least two options. First, we can allow expansions downward only with a fixed width. Second, we can allow any width by using frames if it is impossible to make a restriction on the width of the document to be embedded.

2.3. Examples

The virtual document concept can be best illustrated with examples. Fig. 1 shows an example where the document labeled as VD corresponds to the hub containing two E-links and one R-link, which has been instantiated for a display. In this example, the anchor for the R-link has been activated. The documents labeled as PD represent ordinary documents that may contain multi-media materials. If the PD1 and PD4 are both documents in a CD ROM, the R-link between the anchor “ROSE” in the PD1 and the destination PD4 would be a link between two read-only documents. It should be noted that one of the E-links is partial in that it points to a part of a physical document rather than to the whole document as in HTML documents.

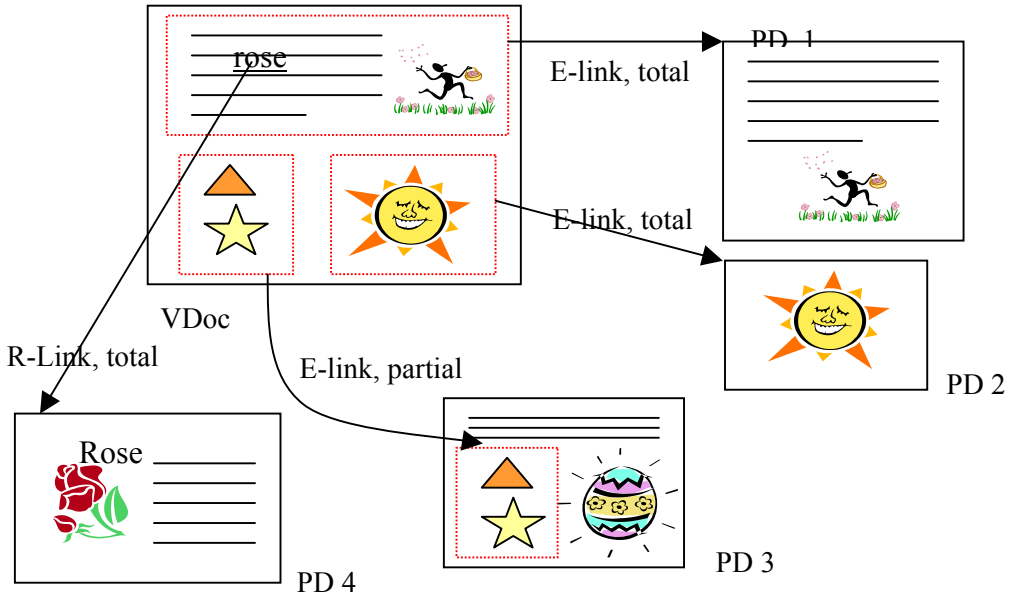

Fig. 1. An example of an instantiated virtual document with E-links and an R-link

Another example in Fig. 2 shows a case with one-to-many relationships for both E- and R-links. For the R-link, users can be given a chance to choose one of the three items as in the Figure, but other presentation methods are possible. Likewise E-link cases can be handled in different ways. The Figure shows just one method by which the E-link is instantiated to display a frame consisting of the four parts that correspond to the four physical documents.

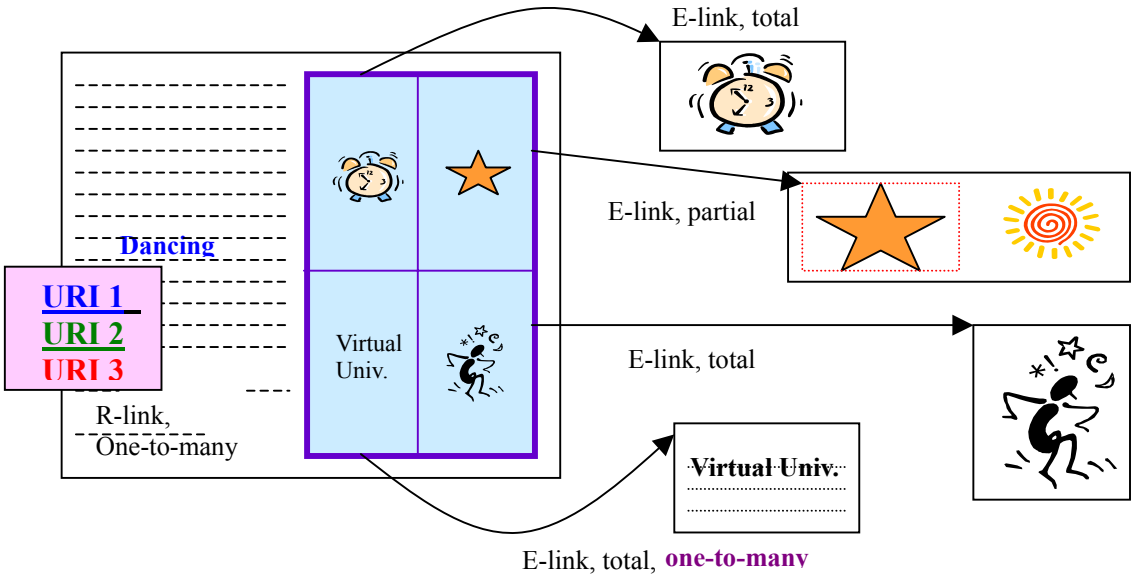

Figure 2. A virtual document with one-to-many relationships

3. A Digital Library Application

While the virtual document concept and its architecture can be used in general digital library (DL) applications, our current interest lies in its application for education because there are many benefits we can obtain. In this section, we first introduce a general-purpose yet education-oriented digital library architecture on which the virtual document concept is realized. Then we discuss how this architecture together with the virtual document concept can be used in educational environments.

3.1. Digital Library Architecture

We propose a digital library architecture where two versions of DL software exist: a regular full-fledged version and a light version. The regular version is built for public use, providing full service as a DL server, and the light version for personal use. The basic functions of the both versions are similar, but the light version has features for building and using personal library. The light version is designed to be installed at a low cost so that people can install it on their PC's and use it to connect to public DL servers as well as manage their own libraries. Both regular and light versions are composed of the following: a user agent (UA), a retrieval server (RS), a storage server (SS), and a link server (LS). A user client (UC) can be connected to either a light or regular version so that the light version can be operated alone using its own data set. However, it should be connected to a regular DL server for services beyond the personal collection.

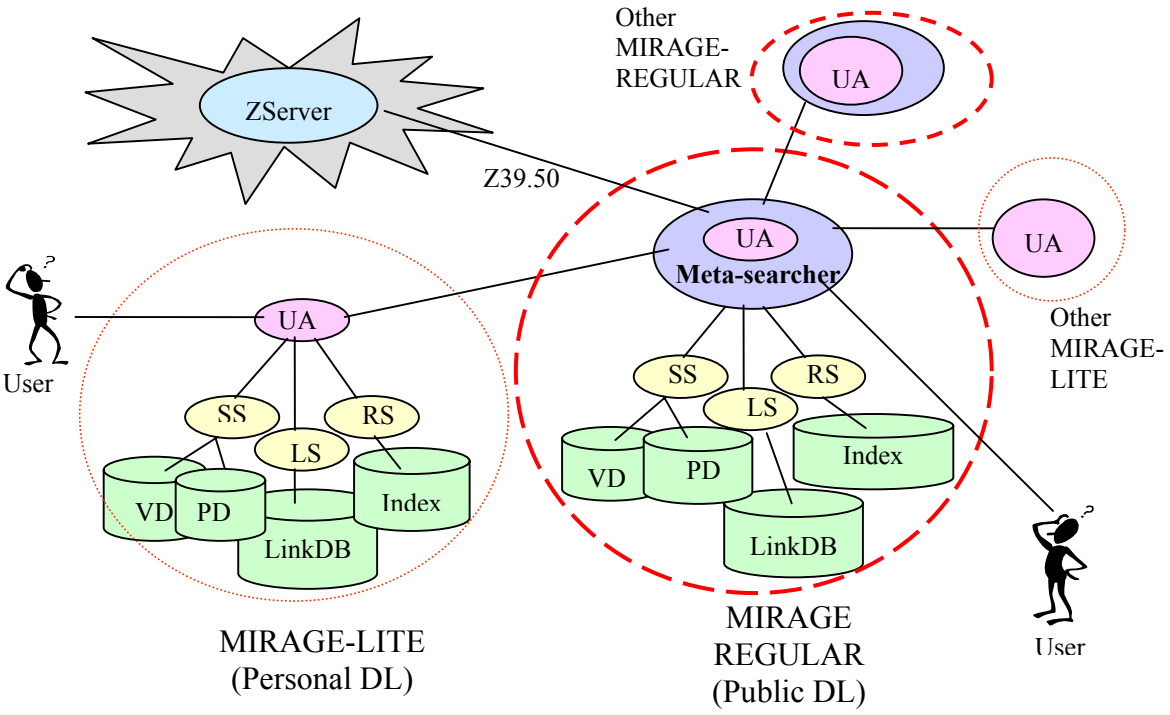

Fig. 3. The MIRAGE Digital Library Environment

The UA plays the key role in the DL system. It receives queries from a user client and returns the results to the users. For this task, it requests the RS to retrieve information for user's query, the SS to transmit the contents of documents, and the LS to send the link information. The UA also has a meta-search function that can communicate with UAs in other DL systems and search for information at remote servers. Z39.50 is used as the standard protocol among the MIRAGE DL systems as well as between a MIRAGE system and external systems for interoperability.

The RS has a set of indices for the documents the DL system has. At the request from the UA, the RS retrieves documents, creates a result set, and sends it to the UA. It is unique in that it ranks documents based not only on the similarity between a query and a document but also on the links found in documents. Another unique aspect of the RS is that it can retrieve composite documents at various granularity levels.

The SS stores and manages both physical and virtual documents. At the request of the UA, the SS transmits a document to the UA so that it can be displayed at the user's client. When a virtual document is instantiated, the user may wish to permanently store it as a physical document. The SS is also responsible for making sure that a single copy for each document be maintained in the database and that documents are modified only within itself. Right management should be done in this server, too.

The LS handles link information in the documents stored in SS. Each user can create and manage its own links independently from other users. The ownership of links can be maintained by attribute of links. There are many benefits we can gain from links. First, we can create links on read only documents like CD titles, video, audio, and documents owned by other persons. Second, many semantically oriented operations are possible.

3.2. A digital library system for education

As the Internet gets popular and available to the general pubic, it has become a new infra-structure for distant education in the cyber space, for which DLs are an essential component. We envision cyber education as the main application area for the virtual document concept and the related DL system architecture. Our DL architecture consisting of distributed servers and an agent is along the idea in the agent-based digital library project of University of Michigan, whose targeted application is high school education [Bur97], as well as the architectures proposed by CNRI [Lei98]. Another effort directly related to our approach is the DL project by University of California at Berkeley, where the concept of multivalent documents (MVD) was developed. Our

virtual document concept was originally inspired by the MVD concept although its intended use and the architecture are quite different.

The environment we aim at is assumed to have the following characteristics:

- There are two groups, teachers and students, interacting together through a network and working on the same subject area.
- A teacher creates materials from her own resources and from DLs, stores them in her own personal DL and makes them accessible by qualified students.
- Students are given assignments that require integration of materials from the teacher, their own private resources, and DLs. Term papers created as such may share the same materials among students, especially those provided by the teacher, although their compositions may differ from each other.
- Teachers and students both use a variety of multimedia materials so that their term papers can be "played" on a display.
- Instructional materials are dispersed in DLs worldwide.

The virtual document concept and the DL architecture are both geared toward providing necessary functionality for various tasks in the aforementioned environment. The user, either a student or teacher, would desire to search for necessary materials from her own DL and/or from public DLs and create a document by organizing them with a multimedia authoring tool designed for this environment. The documents to be composed would contain many links to the materials stored in remote as well as local servers, not the materials themselves. The user would then make it available somewhere in the cyber space to fulfill the purpose (e.g. a term paper or an instructional material). At a later time, the consumer of the material would bring up the document and wish to *instantiate* the document so that all the destination parts of the E-links in the document would be actually transmitted from the servers, embedded to the right places, and displayed finally.

4. **Implementation issues**

DL systems that support virtual documents need to process them efficiently while preserving the merits offered to end users. As such, the internal representation of the virtual documents is a critical issue, for which we considered the following:

- **Expressiveness**: All the functionality of the virtual document concept should be expressed in the representation. The meaning and structure of a virtual document

should be defined precisely and interpreted unambiguously.

- **Processing Efficiency**: A system that supports virtual documents should process them efficiently. It should be easy to parse a virtual document, to make its internal structure at run-time, and to convert it into a physical document if necessary.
- **Accessibility**: Virtual documents should be accessible by various tools available in the Internet so that they can be interpreted and presented by commercial Internet browsers such as the Netscape browser or Internet Explorer.

A natural choice, HTML, was abandoned quickly because it is not flexible enough to express the functionality we need with the virtual document concept although it has become a de facto standard for Internet documents. Another choice, our own representation language, was also abandoned because it would not satisfy the second and third criteria. It would be possible to create a representation language that will allow for the functionality of the virtual document concept. However, documents in such a language would require an extra processing for transforming them to a form with which they can be viewed in an available browser and other Internet-related tools.

We chose XML for representation of virtual documents. XML has been prepared and now recommended as a standard for the next generation markup language for Internet documents by W3C [W3C98]. Among them, XML, XLink, and XPointer are primary standards used for XML documents. XML itself is the most basic standard to define its syntax. XLink and XPointer have been designed for link elements and addressing mechanism. DOM is for a run-time processing model of XML documents and defines the user interface to XML documents. We believe XML satisfies all the three criteria mentioned above.

4.1. The DTD for Virtual Documents.

In order to represent virtual documents using XML, we need a DTD that defines the structure of hubs of virtual documents. In Figure 4 that shows our DTD, we use Arial typeface letters to denote the DTD lines and the DTD elements.

The root element is Hub that consists of three elements: RLinkSet for a set of referential links, ELinkSeq for a sequence of embedding links, and Metadata for metadata of the virtual document. A referential link can be an RLink of the RLinkSet.

An RLink consists of one Source anchor followed by one or more Destination anchors. The Source has an attribute for generic referential links, is_generic. If the Source is generic, is_generic has the value ‘YES’ and the generic expression is given as a value of href whose syntax follows XPointer. An ELink

```
<!-- DTD for VDoc. 1999-10-20 -->

<!ELEMENT   VDocHub        (ELinkSeq, RLinkSet, Metadata) >

<!-- Embedding Links   -->
<!ELEMENT   ELinkSeq       (ELink)+ >
<!ATTLIST   ELinkSeq
            role           CDATA          #IMPLIED
            title          CDATA          #IMPLIED >
<!ELEMENT   ELink                         EMPTY >
<!ATTLIST   ELink
            href           CDATA          #REQUIRED
            role           CDATA          #IMPLIED
            title          CDATA          #IMPLIED
            actuatedefault (user|auto)    "user"
            atuoDelete     (NO|YES)       "NO" >

<!-- Reference Links -->
<!ELEMENT    RlinkSet      (RLink*) >
<!ELEMENT   RLink          (Source, Destination+) >
<!ATTLIST   RLink
            role           CDATA          #IMPLIED
            title          CDATA          #IMPLIED
            showdefault    (new|parsed|replace) "replace"
            actuatedefault (user|auto)    "user" >
<!ELEMENT   Source         EMPTY >
<!ATTLIST   Source
            role           CDATA          #IMPLIED
            title          CDATA          #IMPLIED
            href           CDATA          #REQUIRED
            Is_generic     (NO|YES)       "NO"
            GenericExpr    CDATA          #IMPLIED
            autoDelete     (NO|YES)       "NO" >
<!ELEMENT   Destination    EMPTY >
<!ATTLIST   Destination
            role           CDATA          #IMPLIED
            href           CDATA          #REQUIRED
            title          CDATA          #IMPLIED
            autoDelete     (NO|YES)       "NO" >

<!--   Meta-data / Dublin Core type -->
<!ELEMENT   Metadata
            (DC_TITLE, DC_CREATOR, DC_SUBJECT,
            DC_DESCRIPTION, DC_PUBLISHER, DC_CONTRIBUTOR,
            DC_TYPE, DC_DATE, DC_FROMAT,
            DC_IDENTIFIER,   DC_SOURCE, DC_LAGUAGE,
            DC_RELATION,                DC_COVERAGE,
DC_RIGHTS) >
<!ELEMENT   DC_TITLE       EMPTY >
<!ATTLIST   DC_TITLE
            value          CDATA          #IMPLIED >
<!ELEMENT   DC_CREATOR     EMPTY >
<!ATTLIST   DC_CREATOR
            value          CDATA          #IMPLIED >

            ......

<!ELEMENT   DC_RIGHTS      EMPTY >
<!ATTLIST   DC_RIGHTS
            value          CDATA          #IMPLIED >
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->
```

Figure 4. DTD for Virtual Documents

describes the destination of an Embedding link. The source is the ELink itself. The Metadata, the last element of the Hub, defines 15 items of Dublin Core [Bak98] in order to describe the metadata of virtual documents.

4.2. Authoring and Browsing of Virtual Documents

The authoring tool creates a new virtual document and stores it into the virtual document set. Since a virtual document is an XML document, any XML browser can present it directly on the screen. In our effort to develop an authoring tool, we consider the following characteristics and merits of virtual documents.

- We need to provide a method of taking Internet documents and easily cutting and pasting them, which will depend on the type of the documents.
- The user should be able to created links easily. In particular, it is important to provide a convenient method for creating referential links on the read-only documents, one-to-many referential links, or generic links.
- Users should be able to express the layout and style information in an easy way. Our authoring tool is required to support at least the style and layout mechanism of HTML, CSS2 and XSL.

4.3. Other implementation issues

One of the important features in virtual documents is the global generic link that can be considered to be generic links owned by all the virtual documents. However, if every virtual document keeps a physical copy of global generic links, this will induce a consistency problem with an update on global generic links. This problem can be resolved by maintaining only one set of global generic links as an XML subdocument. This document has a unique name on a digital library system. The RLinkSet of every virtual document only specifies this unique subdocument to be included.

Our DL system maintains a link database. This will help applications using virtual documents (e.g. a search engine that take advantage of links) to efficiently process the links that are included in the virtual documents. The link database keeps all the links in all virtual documents. Whenever there is a change in virtual document links, it is reflected on the link database by the link server. Global generic links are also stored into the link database.

5. **Conclusion**

We have proposed a new document architecture designed for digital library applications in education. The virtual document concept allows for easy creation of new documents within a digital library setting and efficient manipulation of them. The new document architecture is being implemented using XML so that we can achieve expressiveness, efficiency, and accessibility. In addition, we have designed our own digital library architecture that takes into account the characteristics of virtual documents and their applications in educational environment. We are in the process of implementing a prototype and expect to have it up and running in a few months.

References

[Bak98] Baker, T. (1998). “Language for Dublin Core,” D-lib Magazine, December.

[Bur97] Burfree et al. (1997). “The agent architecture of the University of Michigan digital library,” IEEE Computer.

[Lag96] Lagoze, C., Lynch, C., & Daniel, R. (1996). “The Warwick framework: a container architecture for diverse sets of metadata,” D-Lib Magazine, July.

[Lei98] Leiner (1998). “The NCSTRL approach to open architecture for the confederated digital libraries.”

[Pay98] Payette, S., & Lagoze, C. (1998). “Flexible and Extensible Digital Object and Repository Architecture (FEDORA),” Proc. of the 2nd European Conference on Digital Libraries, Heraklion, Greece, Sept.

[Phe96] Phelps, t. & R. Wilensky (1996). “Toward active, extensible, networked

documents: multivalent architecture and applications," in Proc. of 1st ACM International Conference on Digital Libraries, Bethesda, Maryland, March.