제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

Design of Web Content Model

Onytra Abbass, Heung-Seo Koo
Dept. of Computer and Information Engineering, Chongju University
e-mail : onytra@hotmail.com, hskoo@chongju.ac.kr

웹 컨텐트 저장소

Onytra Abbass, 구흥서
청주대학교 컴퓨터정보공학과

Abstract

Managing semistructured data needs fine granularity such as markup elements. XML has major effect in managing web content, it enables content reusability, enriches information with metadata, ensures valid document links, etc. We introduce our content model as an integrated work which handles content objects as controllable units. The paper concerns on modeling news site and how the content is classified due to the site structure, aggregated content and reusability. The model stores instance XML document into relation database using fragmentation strategy.

1. Introduction

Managing web content concentrates on the textual representation when processing and analyzing documents[9]. We have further insight towards content management including sharing and content reuse, integration with operational databases as well as creating database schema automatically to facilitate dynamic content management.

In order to achieve those aspects we propose a design of web content model, which is based on relational database system and XML technology, with wide use of content granularity according to your content properties. Dividing your content into small chunks put them under your control. There is interference between the chunks, XML fragmentation strategy, XML schema and final layout of your site. XML schema ensures that your content is correct and the instance document is well-formed and valid. An adaptable fragmentation is a flexible method to store XML documents (mostly virtual) in relational databases. Site layout assists you in determining the objects granularity, for example, in the news site you can determine content classification relying on static elements that forms the main page, the body of the site which consist of events list which needs regularly updates, the vote and advertisements which need irregular changes, etc.

Our repository model consists of file system which may contain images, media, and static HTML files. Relational database stores XML documents and file reference using fragmentation algorithm.

The rest of paper is organized as follows: Section 2 proposes the related works, including the concepts of CMS, the term of Content model as a set of components and elements, and the relationship between these elements. Section 3 proposes the design of web site content model. We conclude our paper in section 4.

2. Background and Related works

2. 1 Content Model

Content model is a basic part of your repository. This model establishes how each table in the database is constructed and how it relates to the other tables in the database. Since our model will rely on XML, we will use XML schema to establish how each element in the XML file is constructed and how it relates to the other elements in the file[1]. In order to define content model, you have to declare the component classes, the elements included in each component class, and the access structures in which each component class and instance participate.

2.1.1 Component class

Granularity is the level to which a specific chunk or unit of information is defined. The class establishes the definition of a component of a certain type. A component class includes the name of the component class, the elements that it consists of, and any rules that you want to establish about how to create such components. There are many points to consider in determining content granularity. Is the document real or virtual document? Is it grammatically described, such as a paragraph, sentence, or word of text?[2] What is the structure of document template? And how the content is tagged?

In news site you can create classes depending on the content type (text, image, media), the frequently of the updated content (regular changes, irregular changes, and static content), which content represents the bold titles of the site and which represent the body text for each events. The following table shows the characteristics which you apply to your component class.

<Table 1> Component class

ID	Name	Description	Reuse	Metadata
Unique ID	memorable name for the component class	A brief description of the content type or functionality	Is the elements of the class are needed to reused else where?	Class relevant information

2.1.2 Element Characteristics:

All content elements share a few common characteristics, such as elements categorization, each element is associated with a type, the element may be referenced in other content sections but it is only managed from its own content section, and each element associated with one or more templates.

Beside the body elements which contains web content, you have to recognize the management elements to know the administrative information that you think you may need to capture for this component type (for example, Creation Date, Review Status, Expiration Date, and so on)[1]. The following table shows the element context.

<Table 2> Elements declaration[3]

Element name	Component use	Type	Localities	Localization method
Element name	ID of whether the component element is classes that used in body, include this for element management, or both	The field type of the element	The localities where we will represent this element, differently from primary locality (body, mgmt)	Element variants, local elements, local components or non of all

2.2 DyCE

Although DyCE (Dynamic Content Emulator) system[13] is for serving and caching dynamic content, but it uses content objects. This system models the content objects depending on content reusability, and propose size-based and level-based splitting techniques to infer document objects. They conclude that their content granularity determined highly with content reusability, in contrast, our model have various methods to achieve granularity, we rely on site layout, template structure, content tagging, input forms, shared and reusable content. We avoid the use of both size-based and level-based techniques. In addition we create the relational database schema automatically.

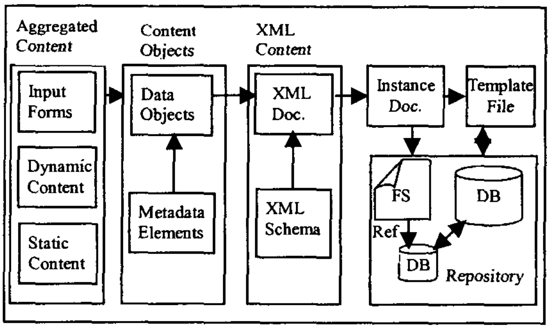
2.3 XCoP

XCoP (XML Content Repository) is a repository system that allows managing content through a set of fragments. The fragment granularity in XCoP is flexibly configurable by the users. Depending on their needs, users may specify, split, and merge fragments at runtime in suitable units they want to share[9]. The system manages the repository of structural information based on ORDMS. Our model concerns on managing unstructured content and based on Relational Database.

3. Model Design

The major components of content model are aggregated content, content objects, XML content, and the relational database schema (See figure 1). A content scheme that is not totally flexible in granularity will inevitably become inappropriate for certain situations. The flexibility enables addition or deletion of components, modification of component behavior, and addition, modification, or deletion attributes. The content model of an XML element-type is much more flexible by an XML Schema, choices between alternative contents, variable numbers of repetitions, and mixed content (subelements mixed with text). So managing XML elements is at the top of our model.

We will use a news site as sample of dynamic content needs to be aggregated, classified, and stored to be ready on scheduled or on demand publishing. It has rich content that will be provided to its web users in an easily-accessible, continuously updated and enjoyable form.

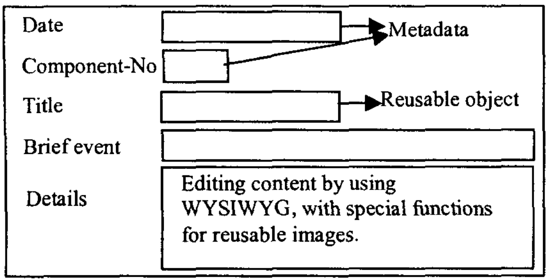


(Figure 1) Web Content Model

3.1 Aggregated content

Aggregation is the process of bringing disparate information sources into one overall structure. This mechanism is related to the content collection part of CMS, but it supports the interaction between the collection and management system. The main advantage of content aggregation is declaring of the content type, resources, and other information. This information represents the metadata which is necessary in identifying and classifying the content. Figure 2 shows a sample of an input form used in gathering news site event.

We detect that date, title, brief event, and component-no (subtemplate ID) are stored as metadata content. Image content is stored in separate table. When we store the event content we add an element which contains a reference to image table and a set of attributes declare image properties.



Editing content by using WYSIWYG, with special functions for reusable images.

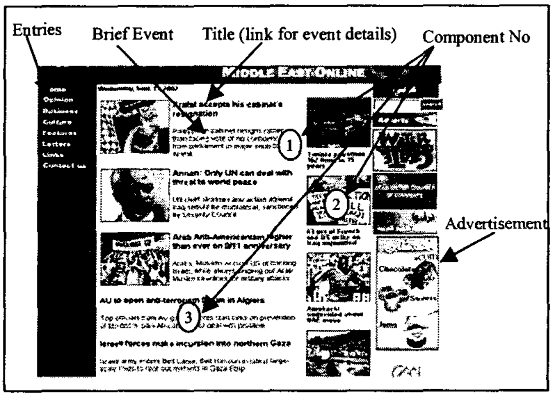
(Figure2) sample of input form (editing event)

3.2 Content Objects

Content object is the heart of our model, it classify and package your content. The process of classification must be governed by useful and meaningful units. There is no restriction for the unit size. The term of object is varying from very small unit to entire document. Content granularity must be available at any level; component size must be dependent on the requirements of the information[4]. But you have to consider the functionality and the storage mechanism of the elements.

The declaration of the parent/child relationship assists you accurately track of all uses of a single content element; separate the content itself from its uses, acquire content, and maximize storage efficiency[5]. The same content can be used in different views, hierarchies, and configurations without duplication. Element locality enable you easily apply the security model to any content object, and determine the inheritance policy. This mechanism is widely used in Zope system. For more information about Components and elements see section 2 "Content Model".

Content is classified into four basic types of objects. File and image objects contain textual and image content respectively. Temporary objects are stored in RAM rather than other storages, so they are suitable for storing small data. In addition, they support the process of rollback and transaction. Properties objects are used to identify your content. It is very useful in providing metadata such as author, title, etc.



(Figure 3) middle-east-online site structure

Dynamic content need efforts to locate each element. The metadata and the expected layout will assist you to determine the type and approximately the size of the content, besides the other information which you can aggregate from different resources. Figure 3 shows site layout which helps in defining database schema.

We automatically create table schema for storing image content (imageID, eventID, content). XML instance document (well-formed) is created containing event elements. Here the use of relational database is appropriate because the resulting database schema doesn't contain numerous tables which need a lot of join processes when retrieving XML elements.

3.3 XML schema

XML has dramatic adoption and ability to model structured, unstructured and semi-structured data. XML Schemas can be created to control which elements can be used in a particular context. Besides inheritance among content model types, span through multiple documents, and improves data interchange, it controls the type and values allowed in a particular context, and define rules for grouping the child elements to form a parent.

For example, a newspaper page has one or more Article elements placed in a random way. The rule for identifying an article element is that there should be a Heading element which has text of larger font size followed by Body element which has text of smaller font size. Following this rule, the segmentation module try to group together text blocks of large font size followed by text blocks of small font size (aligned horizontally) to form an Article element[4].

In particular, XML schema is not required at every point in the information's processing cycle[6]. When you think about making your content useful and make it valuable across your entire enterprise, you will find no way to violate the uses of the XML schema. It is an extremely powerful, yet flexible document definition language that can provide controls over not only element and attribute existence, content, and order, but over specific data types, when elements can be used, and how attributes can be used[7].

3.4. Relational database schema

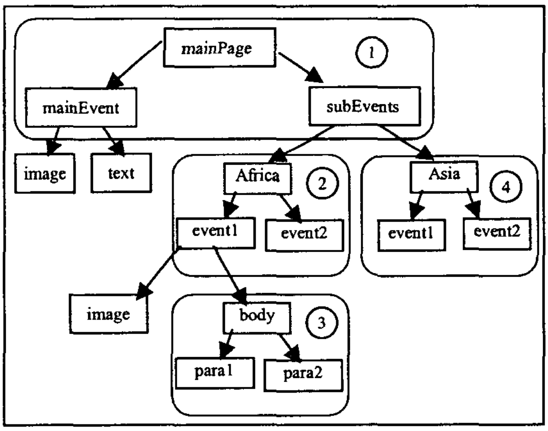
A flexible method to store XML documents in relational databases is presented that is based on an adaptable fragmentation. Different fragmentation strategies depending on the specific access and query requirements can be applied to the same XML documents. The important step in modeling content is creating your database schema and determining the relationships between the tables. The user or database administrator can decide how to store XML elements in relational tables. Appropriate relational schemata can also be derived automatically from a given XML schema[8].

3.4.1 Fragmentation strategies

A fragmentation strategy concentrates on how a set of elements of the XML document can be combined to specify the root elements of the fragments. It determines which element must be stored in separate fragments. The strategy must be built on some preconditions. According to these conditions, you can easily recognize which element satisfy or qualify as a root element of a fragment. Strategies can also be based on the nesting depth, i.e., how many levels of nested elements are stored in a fragment[10]. For the news site we will consider the fragmentation strategy which relies on preconditions to locate the elements. Figure 4 shows a sample of news site XML Document. Figure 5 shows the process of fragmentation.

```
<mainPage>
   <mainEvent>
      <image> reference to event image</image>
      <text> the event details </text>
   </mainEvent>
   <subEvent>
      <Africa>
         <event1>
            <image> reference to event image</image>
            <body>
               <para1> first paragraph </para1>
               <para2> second paragraph </para2>
            </body>
         </event1>
      </Africa>
      <Asia>
         <event1/>
         <event2/>
      </Asia>
   </subEvent>
<mainPage/>
```

(Figure 4) Sample XML Document



(Figure 5) Fragmentation Process

```
<mainPage>
   <mainEvent>
      <image> reference to event image </image>
      <text> the event details </text>
   </mainEvent>
      <subEvent>
         <Africa fragmentId=”2”>
         <Africa fragmentId=”2”>
      </subEvent>
<mainPage>
```

(Figure 6) XML Represent the Fragments in Figure 4

To store the fragments we use a relational schema consisting of the following three tables[10]:
fragment(id, tag, xml)
frag_attribute(id, name, value)
child(parId, childId, pos)

Your can restore XML content by using Query statement and replace each fragment with its XML Content.

<Table 3> Storing fragments in relational database

Id	Tag	XML Content
1	mainPage	Figure 6
2	Africa	<Africa> <event1 fragmentId=”3”> <event2/> </Africa>
3	body	<body> <para1/> <para2/> </body>
4	Asia	<Asia> <event1/> <event2/> </Asia>

4. Conclusion

Web content is varying enough that leads to difficulties in storing it. The content needs to be divided into components that share some characteristics, for each component we have to define its classes and elements and the relationships among them. We presented a model that analysis the site layout and determine grouping objects, and surround them with needed metadata to form an instance XML document. We paid full attention to aggregated content and it properties. Virtual XML document which contains events content (news site) is stored using fragmentation strategy. Although many researchers realize that object databases and Native XML databases are efficient storages for XML document, we found that relation database is the most appropriate storage for our model.

References:

[1] Content Management Bible-Bob Boiko - ©2001 - Hungry Minds Inc -ISBN: 0-7645-4862-X

[2] XML: Text & Context, http://www.webreview.com/1999/02_05/webauthors/02_05_99_1.shtml, 1999

[3] CMS Metatorial Planner, http://metatorial.com/CMS-Metatorial-Planner-Eval-Version.pdf, 2002

[4] The Ten Commandments of Content Management in a Database, http://www.gca.org/papers/xmleurope2001/papers/html/sid-02-3.html, 2000

[5] A Model Guided Document Image Analysis Scheme, http://www.cise.ufl.edu/~nvohra/research/icdar2001.pdf, 2001

[6] Structure rules! , 1999 http://mitpress.mit.edu/journals/MLANG/ensign.pdf

[7] Using Schemas, 2001 , http://www.perfectxml.com/om/IntroSch.PDF

[8] An approach to the model-based fragmentation and relational storage of XML-documents, http://daisy.fmi.uni-passau.de/papers/S01/S01.pdf

[9] XML Content Management based on Object-Relational Database Technology, 2000, http://citeseer.nj.nec.com/surjanto00xml.html

[10] Data Modeling and Relational Storage of XML-based Teachware , http://www.tu-chemnitz.de/informatik/webdb/022.pdf

[11] Modeling Object Characteristics of Dynamic Web Content, 2002, http://www.cs.nyu.edu/~weisong/papers/globecom02.pdf