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

Design of Templating System for Web Publication

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

웹 출판을 위한 템플릿 시스템의 설계

Hisham Abdallah, 구흥서
청주대학교 컴퓨터정보공학과

Abstract

This paper presents a well-designed templating system for CMS web Publication using XML/XSL technology. The primary motivation is the need of Web CMS to separate content from layout and logic. Our system provides GUI XSLT editor (x-editor) to create and modify XSLT stylesheet documents easily. These documents are used to add "layout" and "look and feel" information to XML document which contains content and functionality. The modified XML document is processed by XML-template engine to produce dynamic or static web sites.

1. Introduction

Modern web sites which managed by CMS require highly flexible and dynamic content. Users nowadays expect a web site to offer interactive and up-to-date content. Web Content Management System satisfies these features by a well-designed templating system and editorial system.

A well-designed templating system gives a web site a number of properties. It fosters consistent visual design, an ability to be easily updated with fresh content, separation layout, content and logic -the most important property- to facilitate changes by many developers concurrently and an efficient run-time architecture that lets the web site perform well under load.

Editor offer graphical environment for user to interact with system easily and not required experiences in programmer languages to use it. Templating system needs tool appropriate for designers to help them to generate changeable and flexible templates for pages "layout" and "look and feel" easily.

There is more than one technology to build template system. Model-View-Controller (MVC) Pattern known as model2 (such as Java Server Pages (JSP v0.92)[3] achieves separate business logic (Controller) from content (model) and layout (view) but it is complex.

Another technology is the World Wide Web Consortium's Extensible Markup Language (XML)[8] along the Extensible Stylesheet Language (XSL)[9] aim at the solving the layout and content separation only.

This paper describes our ongoing work to design a well-designed templating system using a layered approach. The content (information, metadata and functionality) is represented in XML document. For "layout" and "look and feel" XSLT stylesheets which generated by GUI template editor (X-Editor) are added to the XML document. XML-template engine processes the XML document to generate last output (HTML for static sites and Java code for dynamic).

The rest of paper organized as follows: In section2, we discuss CMS concepts. We present our templating system architecture and our system template mechanism in section3 and section4. In section5 we give an overview of the related works and summarize our work in section6.

2. Background

2.1 Content Management System

The field of content management covers already a large area of information technology [digital asset management (DAM), document management (DM), knowledge management (KM), software configuration/source code management (SCM), and digital rights management (DRM), web content management (WCM)][6].

There are no universally accepted standards for what content management systems are or do. In the next paragraphs I try to define all necessary terms (content, content management and content management systems).

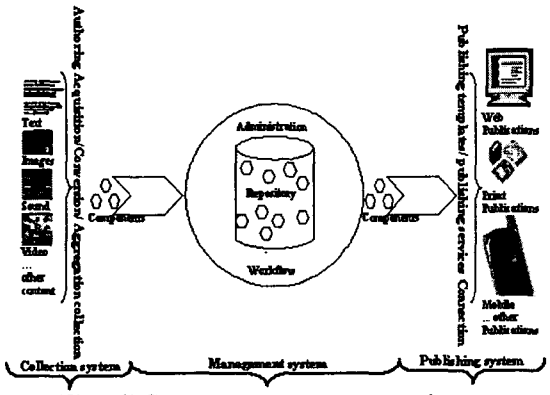
Content is a compromise between the usefulness of data and the richness of information. Content is rich information

This work was partially supported by University-Industry Cooperative Research in Chongju University in 2002.

that you wrap in simple data. The data that surround the information (metadata) is a simplified version of the context and meaning of the information[2]. In general it could be any kind of use full information such as images, graphics, sound, video or text made out of components that categorized, stored, and published based on metadata.

Content management is effectively collecting, managing, and making information available in targeted publications[1].

A content management system (CMS) is software to complete successfully content management. This software enables you to separate the process of site design from the process of adding/updating content. As shown in Figure1 CMS major parts are a collection system, which is responsible for creating or acquiring information from existing sources, and convert it into content components. A management system, which is a sort of database, stores these components. A publication system draws components out of the management system and turns them into publications.



(Figure1) Content management system major parts

2.2 Publication system

Any CMS should have a mechanism allowing it to make content available to target publications. A publishing system uses templates and publishing services (web application) to produce Web sites as well as other kinds of publications[1] see figure1.

In web publication content can be deployed on the Web server for access by the application in two general ways static production and dynamic publication [4]. With static production, CMS uses page templates to assemble the content into final HTML. Static product is ideal for pages that will not change since they can be served more quickly and can easily be cached in memory, providing a higher performance experience for end users. In case of dynamic publication, the HTML pages are generated dynamically by the CMS or Web application based on the page template requested and the content elements defined on it. Content that changes frequently or is personalized for each user requires dynamic production; however, dynamic production requires additional CPU cycles to generate each page, which slows performance. In addition, since the pages don't exist until they are published, they cannot be cached in memory. In practice, most web applications use a combination of techniques to manage the presentation of content to web users. Some pages change little and are created as static HTML, while others are driven by user profiles, time of day, or other factors controlling final presentation.

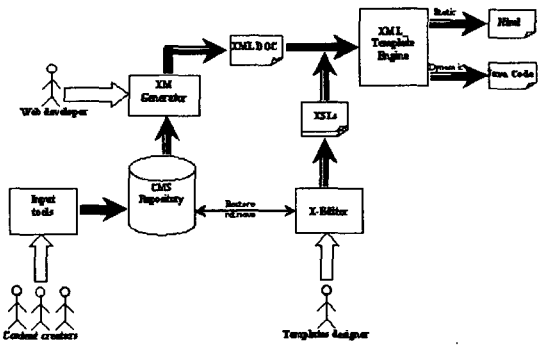
2.3 Templates

The basic idea behind a CMS is to separate the management of content from design. Page designs are stored in templates while the content may be stored CMS repository. When a user requests a web page, the parts are combined to produce requested page[5].

Templates are programs build publications from the content in the repository[1]. Templates include static elements (such as text and media) that are passed directly through to the publication, as well as dynamic elements which are retrieved by publication services (web application).Templates include navigation-building and personalization logic as well as calls to services outside of the CMS that integrate publications into a wider organizational infrastructure .

3. Templating System Architecture

Our Templating System (see figure 2) supports strict LCL separation for static and dynamic pages as well as other requirements for a well-designed templating system; also it supports graphical layout formats through visual template editor, which enables template designers to achieve their job easily. In the rest of this section we will present our Templating System.



(Figure 2) Templating system for web publication

XML DOC

It is well formed xml Document generated by XML generator from CMS repository. It content especially elements will be parsing by template engine. Theses elements represent content and its structure. The template functionally such as (variables, loops, and database queries) is exploited by using special element in XML DOC by web developer. XML DOC can be based on a document type definition (DTD). CMS repository and XML generator are out of subject of this paper.

X-Editor and XSL

The eXtensible Stylesheet Language (XSL) and the eXtensible Stylesheet Language Transformations (XSLT) are standardized languages for transforming XML documents to a different output form. Using XSL, content saved in an XML format can be transformed into any output media, by applying an XSLT stylesheet.

Writing even the most basic XSLT stylesheets by hand is a truly daunting task, requiring an understanding of XSL elements, the XPath query language, and complicated rules-based document processing models.

The X-Editor (XSLT Editor) automates writing of complex XSLT stylesheets through an intuitive, drag-and-drop user interface. Through a powerful GUI, a user can drag and drop XML data elements corresponding to an XML Schema or DTD, adding descriptive text and presentation tags such as tables, hyperlinks and graphics. The resulting XSLT stylesheet is automatically generated (XSL file). This file is applied to XML DOC to add layout information to the document. Moreover, another XSLT stylesheet can be used to add common layout information "look and feel" like header, footer and navigation or restructure document (see next section). Lets call the new document modified XML DCO. Here is an example for XSL file which maybe generated by x-Editor:

```
<xsl:stylesheet xmlns:xls=...>
 <xsl:template match="/">
  <HTML> ... <BODY>
   <xsl:apply-templates/>
  </BODY></HTML>
 </xsl:template>
 <xsl:template match="heading">
  <H1><xsl:value-of select="."></H1>
 </xsl:template>
 <xsl:template match="center">
  <P ALIGN="CENTER">
    <xsl:value-of select=".">
  </P>
 </xsl:template>
 <xsl:template match="list">
  <OL><xsl:apply-templates/></OL>
 </xsl:template>
 <xsl:template match="list/item">
  <LI><xsl:value-of select="."></LI>
 </xsl:template>
</xsl:stylesheet>
```

Templates are dynamic so templates information is stored in CMS Repository. As I mentioned before, work with X-Editor is easily and not require any knowledge in software word.

XML-Template Engine

It processes the modified XML DCO to generate output using SAX parser[7]. The output either HTML for static page or Java code which can easily be used by servlets depends on input XML DOC.

The XML-Template engine distinguishes between static and dynamic documents. Document is considered static if its all elements can be resolved during processing time; otherwise dynamic document such as document contains dynamic database query element or user-defined variables. In dynamic case, the XML-Template engine directly generates a piece of java classes encapsulating the layout and the functionality of the dynamic XML DOC.

4. Template Mechanism

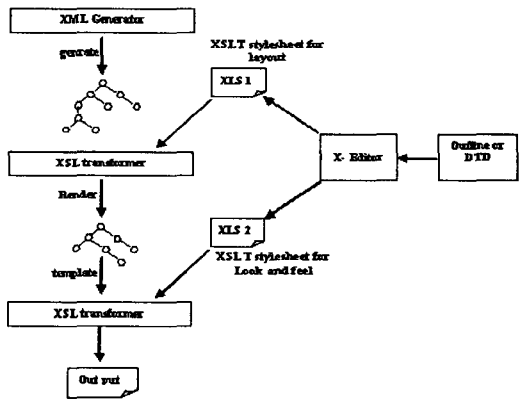
Template is simply XSLT stylesheet (XSL file) used to add "layout" information (rendering content) and "look and feel" information (templating content) to XML DOC. Publishing a piece of content taken from CMS Repository can be seen as a 3-step process show in Figure 3.

The first step generate XML DOC (content, metadata and functionality) this step out of the scope of this paper.

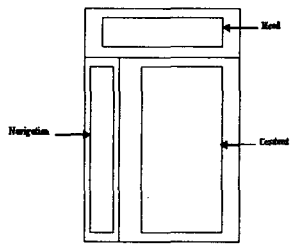
The second step consists in adding layout information (rendering the content), using XSL templates. Such templates should supposedly be defined at the content-type level: "images" should have their own templates, "tables", "articles", and "reports" as well. Thus templating an "article" will be done by calling an "article" template, which in turn should be able to invoke an "image" template if this article contains any. Concretely this can be done by combining two mechanisms:

- an XSL stylesheet can include another one with <xsl:include>
- parameters specifying which sub-templates to invoke may be passed by the original request, and/or by the XML source

The final step consists in adding common look and feel information to the content. This look and feel can include defines the general design of a page (common layout information like header, footer and navigation or restructure document), see Figure 4. This should be done using a global XSL stylesheet, which will setup the overall look and feel of the "page", customize a few dynamic tags using the metadata (for e.g. displaying the date at the top of the page), and insert the piece(s) of rendered content in suitable places.



(Figure 3) XML/XSL workflow



(Figure 4) page common layout

5. Related Works

There are many Content Management System commercial products and open source projects. Of course all this systems have template mechanisms for publishing.

In openCms[10] The XML template mechanism using dynamic page rendering allows a strict separation of content and layout and a unitary display of content for a consistent corporate design. Every template is combined by two parts: the template XML file and an associated Java class. The Java class controls the rendering of the template file and includes dynamic content like navigation or personalization. In simple cases the default template class can be used, for special purposes a customized class is needed. Each page is combined by several templates: The frame template for the corporate design, the content template as a container for the content and a body template for the edited content. Each template can call sub templates to include the rendered output. E.g. the frame template normally calls a navigation template to include the dynamic generated navigation.

Zope[11] have two objects act as web templates DTML objects are Zope's main page templates and Zope Web templates. DTML is a tag-based template language that lets you insert dynamically generated text in HTML document. Zope combines DTML tags and static HTML text to create document template. DTML suffers from a failure to separate presentation, logic, and content .Zope has also Page Templates which use the Template Attribute Language (TAL).TAL is based on XHTML attributes, which can attached to ordinary HTML tags. Not like DTML Page template keep code out of templates, except for structural logic. Both Zope objects template use acquisition to create simple and even complex template structure.

The Cocoon[12] project takes the content and layout separation problem in web sites by using a servlet engine for the real-time application of XSL style sheets to XML files. The Cocoon project proposes XSP (eXtensible Server Pages) technology for providing flexible and layout independent dynamic content in web pages. XSP is completely based on XML/XSL technology and uses XSL tag libraries and associated code generation style sheets (logic sheets) to generate compliable source code. Basic mechanisms for processing XML documents is dispatching based on matchers, generation of XML documents through generators (from content, logic, Relation DB, objects or any combination), transformation (to another XML, objects or any combination) of XML documents through transformers, aggregation of XML documents through aggregators and rendering XML through Serializes.

The most important issue worth to be mentioned here is that templating systems for all above system not support Graphical Layout Formats.

6. Conclusions

The basic idea behind a Content Management System is to separate the management of content from design .This separation accomplish by templates so templates play basic role in CMS.

With clear LCL Separation and other requirement for a well-designed templating system we can build flexible web site which support easy-of-change. The responsibility can be divided among people who response for content, people who response for layout and developers who response for functionality.

Our solution achieves strict separation content from layout and logic and other requirements for a well-designed templating system. The content and its structure represented in well-formed XML document. Logic defines separately in an arbitrary programming language. Layout and "look and feel" information is added to the content defined in the XML document as separate flexible and changeable XSL document which generate by designer easily and cheaply using X-editor.

References

[1] Bob Boiko, Metatorial Services Inc: Consulting & Education on CMS, www.metatorial.com
Content Management Concepts, http://metatorial.com/Papers/cm_concepts.asp
Why Content Management? , http://metatorial.com/Papers/iqpc_singapore.pdf
[2] Content Management Bible-Bob Boiko - ©2001 - Hungry Minds Inc -ISBN: 0-7645-4862-X
[3] Understanding JavaServer Pages Model 2 architecture, http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc_p.html
[4] Content Management Systems, http://www.e-icore.de/pdfs/cms_tools_generic.pdf
[5] Content Management System Evaluation,http://www.atnf.csiro.au/computing/web/cms_eval.html
[6] Diane Dent,Lynn Thorsell: Web Content Management Product Evaluation Considerations:http://www.ceiss.org/shared/papers/Dent_Thorsell.pdf
[7] Simple API for XML SAX parser,http://www.saxproject.org/
[8]. W3C, eXtensible Markup Language (XML), http://www.w3.org/XML/
[9]. W3C, eXtensible Stylesheet Language (XSL), http://www.w3.org/Style/XSL/
[10] OpenCms: Homepage, http://www.opencms.org/opencms/opencms/index.html
[11] Zope, http://zope.org/
[12] Apache Cocoon, http://xml.apache.org/cocoon/