Study of A Content Oriented Web Architectural Model

Li Qingshan
Software Engineering Institute, Xidian University, Xiän , Shaan'xi, 710071, China
liqingshan@sei.xidian.edu.cn

ABSTRACT

Many larger and complex web applications, such as e-commerce, ASP, portals, are content oriented, in which content is the critical elements and always under constant updates; content organization and presentation generally is required custom-tailored. Developing and maintaining such web applications demands a suitable model that can capture its features. For architectural descriptions address high-level aspects of the systems such as the overall organization, the decomposition into components, and the way the components interact, we first analyze the features of typical content oriented web application framework. In succession, this paper defines the types, structure, semantics, constraints of component and connector in our conceptual model of web content component, within which a web content component is regarded as an independent process unit performing necessary content organizing, processing and presenting functions. In component standpoint, a web application is regarded as a collection of web content components and some other service components providing certain content service. In this paper, our first concern has been to identify, classify, and support a variety of web content components and their connections, serving more effective development and maintenance of web applications.

Keywords: Software Model, Web Application, Web Content Component Model, Component, Connector

1.INTRODUCTION

Web-based application development method has experienced several stages or phases from static information deploying to simple dynamic page generating, and now to more complex functionality information accessing around database at server side. Each step during the evolution is typically a result of the need of bigger size and more complexity of web application logic requirement. In the view of framework of software system, web-based applications using traditional three-tier client/server mainly adopt data centric development model because their large data organization quality and performance are critical to the whole system while complicated functions are at secondary position. However, not all web applications are database extensive applications. Web applications, such as e-commerce, Application Service Providers(ASP), portals, are content oriented and appear more and more important in various web applications requirements. In content oriented applications, content is the main element that both user and web developer care about. Content oriented applications implement content providing, content organizing, content processing and content presenting while data from database is processed by functional modules to provide certain content information and is put at low storage layer. There is a need for suitable model and environment for the analysis, design and maintenance of such systems.

In the view of requirements, many types of content oriented web applications typically have three characteristics as followed compared to general conventional software systems and other web–based systems. First, content, which involves in data and data processing according to appropriate business rules and logics and is provided to end users and intranet client users, is the critical element of the content oriented web applications. Second, these system always provides a web site, in which content is under constant updates and content relations need effectively maintain during the lifecycle of the software. Unlike conventional systems, maintenance, update and future development are major tasks of developing such web application and are extremely important to its vitality. Content oriented web application is always updating its content and functionality in order to suit for constantly changing content requirements. The maintenance is tangled with the redesigning and the latter may happen as frequently as the former[1]. Without a careful design, the web application could quickly become non-maintainable as updating can easily changes the structure of a web application and hence make the original design functionality no longer applicable to the changed structure. The boundary of the development and maintenance stages is no longer as clear as that in conventional software model, and so it is not suitable to develop such systems with traditional concepts and methods. Along the line between several stages of software engineering in web-based application are not as clear as that of traditional software, it is more essential for us to build the system on a model that support engineering methodology of effective web

0-7695-1381-6/01 $10.00 © 2001 IEEE

application development and maintenance. And thirdly, content provided for different kinds of client may be different in content organization, process logic and content presentation. Because web-based applications are meant to be for a wider user base whether within an organization, across a number of organizations or over the internet than traditional applications, web application developer do not necessarily know all the users[2]. In these process environments, the system can remember the history information and useful dynamic data to perform the functionality while it is more sophisticated to control the content description for different users and different application requests using conventional data centric method processing data and providing content.

Component technology has been recognized as a promising and mature solution to software crisis[3].We have make a research on web application model that is about software design and implementation[4], in which a web application is regarded as a collection of components, each having its own functionality and cooperation with others through certain interfaces. Current component technologies used to web application development have provided powerful elements for constructing applications but do not sufficiently considered content providing, organizing and their relations. The architecture of a software system defines that system in terms of components and of interactions among those components. In addition to specifying the structure and topology of the system, the architecture shows the intended correspondence between the system requirements and elements of the constructed system. It can additionally address system-level propertied such as capacity, throughput, consistency, and component compatibility. Generally, a sound basis for software architecture promises benefits for both development and maintenance. Design should benefit from improved abstractions, notations, and analysis. Architectural definitions should help provide good specifications for programming activities. Maintenance should benefit in two ways. First, about half of maintenance effort is dedicated to understanding the system in preparation for making changes; explicit definition of the architecture should reduce this cost. Second, system architectures degrade over times; carrying the architectural definition into maintenance should reduce this tendency toward degradation. For the web application was born with a component oriented nature and content nature, the marriage of these two concepts and technologies will bring a promising prospective for web application and web engineering. From content view, we have developed a practical case relevant to B2B e-commerce based on the integration of content and component[5]. In this paper, we make a deep research on Web Content Component Model(WCCM), which is just the suitable model of web application that can capture the features of development and maintenance of content oriented web applications. Our first concern is to improve the efficiency and effectiveness of web-based application design and maintenance using our web content component model.

2. FRAMEWORK BASED ON WCCM

Software architectures shift the focus of developers from lines-of-code to coarser-grained architectural elements and their overall interconnection structure. While hardware designers recognize a number of distinct design levels each with its own design issues, models, notations, componentry, and analysis techniques[6],different levels of software design require different kinds of components, different ways of composing components, different design issues, and different kinds of reasoning. The vocabulary gap between requirements and programming is substantial. Filling the gap requires better models and notations for the intermediate step.

From WCCM view, a typical content oriented web application consists of a number of Web Content Components(WCC) and auxiliary Web Service Components(WSC). According to the requirement logic, the skeleton of the framework is hierarchical structure and the main relations among WCCs and WSCs are "has" and "use" relationship. We adopt XML as the format for data exchange among components in our model. The conceptual model of content oriented web application framework can be shown in figure 1.

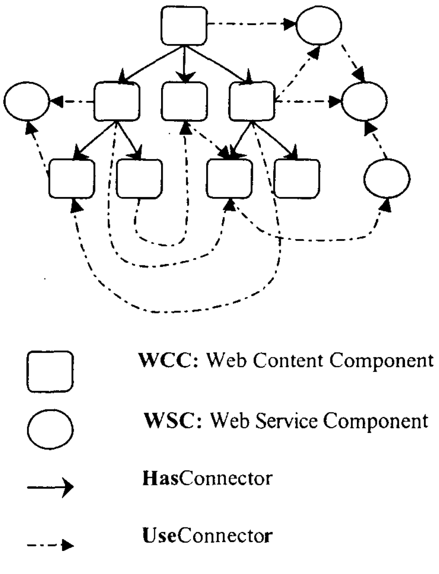

Fig.1 Application Framework based on WCCM

For the characteristic of meta language and nature structured documentation[7], XML(extensible Markup Language) is fast becoming an public industry practical standard for data presentation and data interchange within web applications[8]. Because XML may be easy to write programs that process XML documents, be used straightforward over the Internet, be compatible with existing standards such as SGML and be a meta language, XML and underlying technologies can implement the separation of presentation from content, the separation of business logic from UI logic and data interchange over networks without losing anything in translation. Currently, most typical web-based applications uphold the access and usage of XML model and underlying technologies. Considering the characteristics of content oriented systems and features and functions of XML model described above, XML data format is used for content description and data interchange across components and applications in the WCCM.

With the emphasis on the high level of software development, WCCM is used as a basis for an effective engineering methodology and supporting environment of large and complex web applications development. Based on WCCM, web developer can make software analysis, design and maintenance more coherent because of content and their relations almost have the same structure and represent higher consistency. So such web applications should be developed iterative and of recursion with the support of web content component model and methods.

3.CONCEPTUAL MODEL OF WCCM

Web Content Component

WCCM put emphasis on a mapping of application logic to software design, so the software design is closely relevant to software requirement analysis. In WCCM, Web Content Component(WCC) is an independent logic entity that consists of several functional modules performing content providing, organizing and presenting functions and cooperating with other WCCs and Web Service Components(WSC) through certain interfaces. The content in the WCC includes data, data structure with certain style, data organization suitable to concrete application semantic logic and data presentation. The conceptual model of WCC can be defined below and the structure of WCC may be shown in figure 2.

WCC ::= <{PU, DF, XML_TP, XML_P, PTP, PG},
{CM, DM, MM, Leader, Port}>

A WCC consists of two sections: content manager and interface manager. Content manager involves in Process Unit(PU), Data Filter(DF), XML Template Producer(XML_TP), XML Producer(XML_P), Presentation Template Producer(PTP) and Presentation Generator(PG), which provides several functional modules to perform content organization, process and presentation. Interface manager involves in Context Manager (CM), Data Manager(DM), Maintenance Manager(MM), Leader and Port, which provides several interfaces for the content management in order to exchange information with outer WCCs or WSCs.

Besides the constraints of syntax in the conceptual model of WCC, semantic restricts is also defined in the next.

(1) Content manager

- **PU:** Process Unit performs certain functions such as search, statistic and cooperation, etc, under the control of context data dispatched from leader.
- **DF:** According to certain user information and system dynamic state, data filter can be used to process data,
  organize content and filtrate away irrespective content satisfying the requirements of functionality and organization of information in web application.
- **XML_TP:** In terms of context data and requiring of data from data filter, XML template producer can be used to produce XML template reflecting the relations of content.
- **XML_P:** According to XML template, XML processor transforms the data from DF into XML-tagged data.
- **PTP:** Presentation template producer produces data presentation template according to the requirements of user interface logic and client individuation information, for example, HTML template.
- **PG:** Presentation generator can style xml-tagged data from XML_P for custom-tailored content presentation. XML can separate content from presentation through adding domain-specific published vocabulary to XML-tagged structured data. By published means that the vocabulary has its words defined and the semantics of their use documented. XML is not useful as part of an open, solution-centric world until both XML and vocabulary are married. FO(Formatting Object) represent domain-specific published vocabulary in XSL, which is a stylesheet language for formatting XML documents and styling XML-tagged data according to custom-tailored information in UI logic or business logic of web applications. For example, XML-tagged data can be rendered using transcoding in one of several ways: HTML for standard browsers, DHTML for more modern

browsers, and XML for clients that understand XML.

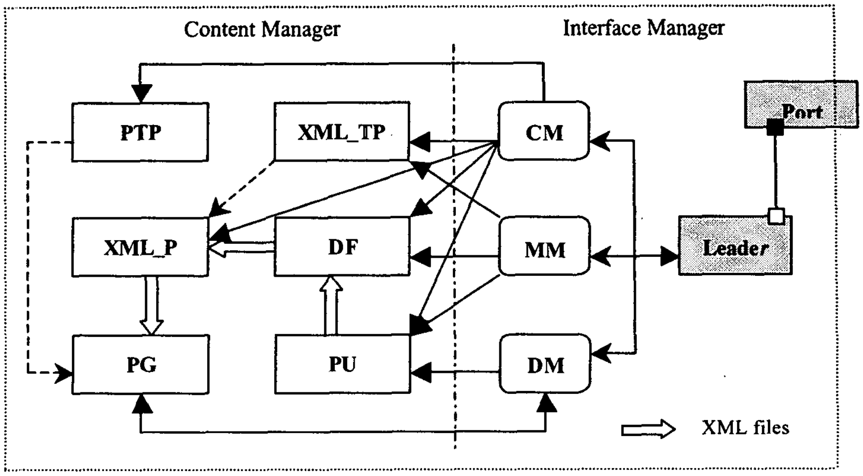

Fig.2 Conceptual Model of Web Content Component

(2) Interface manager

- **CM:** Dispatched from leader, context manager provides context data for WCC to manage content. The context data is a kind of meta data that can control such modules as DF, XML_TP and PU to perform specific content processing and content presenting functions according to user profiles. All the user control data, history data reflecting a certain domain application requirements and session data remembering relevant request information from client side for successive processing effectively can make the system correspondingly deal with user request and give right presentation format based on different context control information.
- **DM:** Dispatched from leader, data manager can complete the bi-directional exchange of xml-tagged data with the outer WCCs or data from database and legacy storages.
- **MM:** Dispatched from leader, maintenance manager may transfer control data about the maintenance request, which involves in the content and the content organization structure, into the certain WSC performing maintenance management functions in the same form as the nature relations among data.
- **Leader:** Leader provides further dispatching of content and metadata, performing the core management of interface of WCC.
- **Port:** Port provides the public exclusive point for communication with various types of connectors. Bound with role of connector, port has specific syntax and semantic constraints.

Web Service Component is inevitable in the web-based application development based on WCCM. The conceptual model of WSC can be defined in the such form: WSC ::= <{FM},{Port}>. FM denotes functional modules providing specific service. WSC perform specific functions such as context data processing, page generating, maintenance implementing and data transferring over different platform and different storage medium. WSC is independent and self-contained, not closely relevant to content process like WCCs, and can communicate with other component through their published interfaces. In WCCM, WSC is such component that can perform supplementary functions to provide certain content and other special manipulations for WCC.

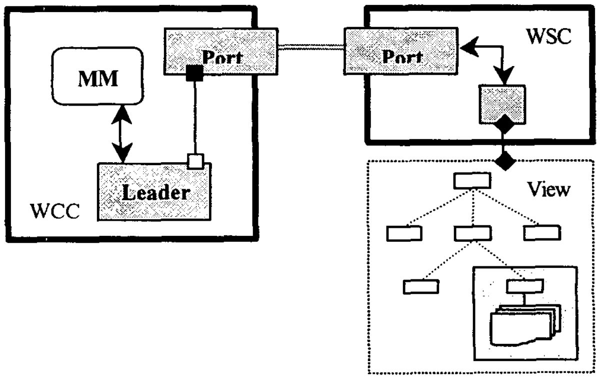

Fig.3 Maintenance Mechanism of Web Content Component

Connector

Connectors mediate interactions among components; that is they establish the rules that govern component interaction and specify any auxiliary implementation mechanism required. During system design, it is important to work with good abstractions for interaction without concern for whether their implementations are localized. In WCCM, there are two types of connectors: HasConnector and UseConnector. The semantic constraints of the connector can be described below.

HasConnector:=<{WCC, WCC}>;
UseConnector:=<{WCC, WCC},{WCC,WSC},{WSC,WSC}>

HasConnector confines that a WCC contains other WCCs providing certain content service according application logic. UseConnector is suitable for the relations among WCCs and WSCs. The semantics of UseConnector means one component can use relevant content that is useful for the component but produced by other components. Compared to the restricts of UseConnector among WCCs and WSCs, the WCC that contains other WCC may has close limit in terms of data structure and logic semantics with each other. The relations among content components reflect the relations of content blocks corresponding to application logic. Communication mechanism through connector may be illustrated in Figure 4.

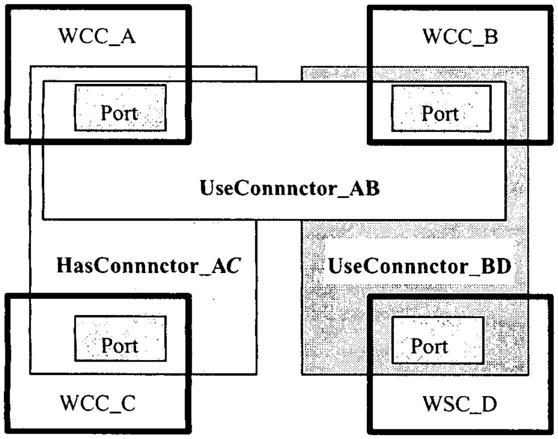

Fig.4 Conceptual Model of Connector

While all the data across WCCs through HasConnector is XML formatted data, the UseConnector may have more abundant semantics. UseConnector performs two main functions, data exchange and data transformation. The WCC can receive XML data from other related component and output XML-tagged data to outer component completing exchange operation of the interface. On the other hand, some WSC can provide other forms of data to WCC based on UseConnector. In this area, based on standard XML protocol and APIs like tree-based DOM, event-based SAX and other interface model like object-oriented data binding model,the connector can transform other forms of data from outer service component to standard XML-tagged data. Specially, all the operation describing above can compete reverse direction data exchange and transformation in the stage of maintenance.

4. CONCLUSION

In the earlier sections we bring forward and discuss thoroughly the structure, type and semantics constraints of web content component model. Utilizing XML as standard and public data format for exchange, it is easy to implement custom-tailored content organization and presentation according domain specific application requirement logic. Based on WCCM, we can get the advantage of the coherence of application logics, design and maintenance in content oriented web application development. Because of the similar structure and content among software design and maintenance based on content component, we could make re-design works more effectively to reduce the maintenance costs and ease the maintenance tasks.

5.REFERENCES

[1] Jeni Li, "The Tangled Web: Designing for Maintenance", Shoecraft,http://ausweb.scu.edu.au/proceedings/shoecraft/paper.htm

[2] R.Debreceny and A.Ellis, Rahardja,(1999) "Designing Interactivity: Who are the users and what are the techniques", Proceedings of AusWeb99 the Fifth Australian World Wide Web Conference, Southern Cross University Press,Lismore,Australia,pp.107-118.

[3] Anders Kristensen, "Developing HTML based Web Applications", 1st International Workshop on Web Engineering WWW7,1998, http://www-uk.hpl.hp.com/people/ak/doc/webe98.html.

[4] Weiquan Zhao and Jian Chen, "CoOWA:A Component Oriented Web Application Model", TOOLS 31 Conference, September 22-25,Nanjing, China, China, pp.316-319, IEEE CS Press,1999.

[5] Qingshan Li, Jian Chen, Ping Chen "Developing an E-commerce Application by Using Content Component Model", TOOLS 3 6 Conference, October 30-November 4, Xi'an, China, pp.275-284, IEEE CS Press,2000.

[6] C.G.Bell and A.Newell, Computer structures: Reading and Examples New York: McGraw-Hill, 1971

[7] J. Bosak, "XML, Java, and the Future of the Web" http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm, 1997

[8] R. Khare and A. Rifkin, "XML: A Door to Automated Web Applications", IEEE Internet Computing, 1(4), July-August 1997, 78-87