



STUDY ON A DISTRIBUTED ARCHITECTURE BASED ON COMPONENT TECHNOLOGIES WITH APPLICATION IN A MANAGEMENT INFORMATION SYSTEM

Jue Wang¹, Qiang Lin² and Sun Zhou^{1,*}

¹Department of Automation, Xiamen University, China

²Department of E-Business, Jimei University, China

Email: zhousun@xmu.edu.cn

Abstract

With the development of Internet/Intranet, more and more traditional applications can be performed on browser, or E-Business trades and other complicated application systems such as MIS, ERP and CRM and so on. In order to satisfy both traditional application and Web application, reduce the cost of system developing repeatedly, increase the reusability of application system and take advantage of the functions of OOA/OOD sufficiently, a new distributed architecture (system solution based on repeated components) comes forth. In this paper, the distributed architecture structure will be discussed in detail. The evolution of traditional application system and Web application system is discussed, especially the advantages and disadvantages of two architectures. The distributed architecture (system solution based on repeated components) is described particularly. Several prevalent component technologies, such as COM+, CORBA and EJB, is also be illustrated. The practice of the technology of reusable component is analyzed and the design of a management system of Science & Technology Articles is presented. The latest distributed technology of Web service based on xml/soap is summarized.

Keywords: distributed system architecture, distributed component technology, web service

1 INTRODCUTION

With the evolution of application system, there are two different system architectures. One is the traditional application system such as MIS and Business Software. Along with its extension, the traditional application system transforms from file-based structure to C/S and then CSS model. At the same time, following the prevalence of Internet/Intranet, the other architecture, Web application system based on browser, comes into being. In early time, Web application only provided functions of looking over or querying static data on browser. But with the development of Internet/Intranet, more and more traditional applications can be performed on browser, or E-Business trades and other complicated application

systems such as MIS, ERP and CRM and so on (Veeraraghavan, Panetta, Agaian, 2015). For a long time, people in the software field always hope to establish a reusable thing and a method of making out reusable thing. Make the strenuous efforts of the engineers/procedures not to vanish in the transfer in time and changes in personnel. From *subroutines, procedures, functions, classes* to *function libraries, class libraries* and then *components* (Zheng, He, Pang, 2014); From *Structure Design(SD), Module Design(MD), Object Oriented Design(OOD)* to *Design Patterns*; From *components technique* to *system architecture* and *framework* in the software engineering.

For the architecture of application system, there primarily are two different system. One is traditional



application system, which is heavily dependent on target operate system platform. At present, there mainly are two platforms Windows-based and Unix/Linux-based. The other is Web-based application system, which is independent of target platform. With the broad application of Internet/Intranet Web-based system is paid more attention.

The evolution of traditional system architecture is described as following:

- (i) File-based: use OS file system to store *applied data*, its weakness is easy to understand.
- (ii) Single model: use simple DBMS to store *applied data*, but the *applied data* cannot be shared.
- (iii) C/S model: *applied data* is stored in large DBMS, but *Presentation* and *Business* are mixed up.
- (iii) CSS model: separate *Presentation*, *Business*, *Data* respectively, decreasing the maintainability as well as increasing repeatability.

Accordingly, the evolution of Web-based system architecture is turned into following stages:

- (i) File oriented: only provide browsing and querying static data using HTML and Client-Script language such as VBScript and JavaScript.
- (ii) Program oriented: offer dynamic browsing and querying using Server-Script such as Asp/Jsp/CGI/ISAPI and so on. But it also encounters the same problem as C/S model in traditional system does.
- (iii) Service oriented: it's namely Web Service we will describe in the following section.

To summarize, traditional system and Web-based system have each merit and weakness. Traditional system has strong power in *presentation*, but increases maintenance in Client-side, especially when there are a lot of Clients. On the other hand, Web-bases system requires caring little about the Client-side maintenance, while it must still struggle to increase the ability in *presentation*.

2 DISTRIBUTED COMPONENT OBJECT MODEL

In general, traditional system is developed first. And then when it comes to Web-based system, it always uses existing Web technique to rewrite *business logic* of existing business system, so the cost of this is obvious. However, is there a architecture which can void this disadvantage by combining traditional system with Web-based system? The answer is so-called *distributed component model*, which firstly encapsulates *business and data logic* in components which are also called *Enterprise Objects*. And then when it comes to develop either kind of the two application systems, the existing components can be reusable in both traditional system and Web-based system. At present, there primarily are three standard component models such as EJB, COM+ and CORBA. In the following, we will make a powerful description between the two choices that businesses have for building distributed application system: EJB (J2EE Object Model, built by Sun Microsystems and other industry players) and COM+ (Windows DNA Object Model, built by Microsoft Corporation).

2.1 EJB: J2EE Object Model

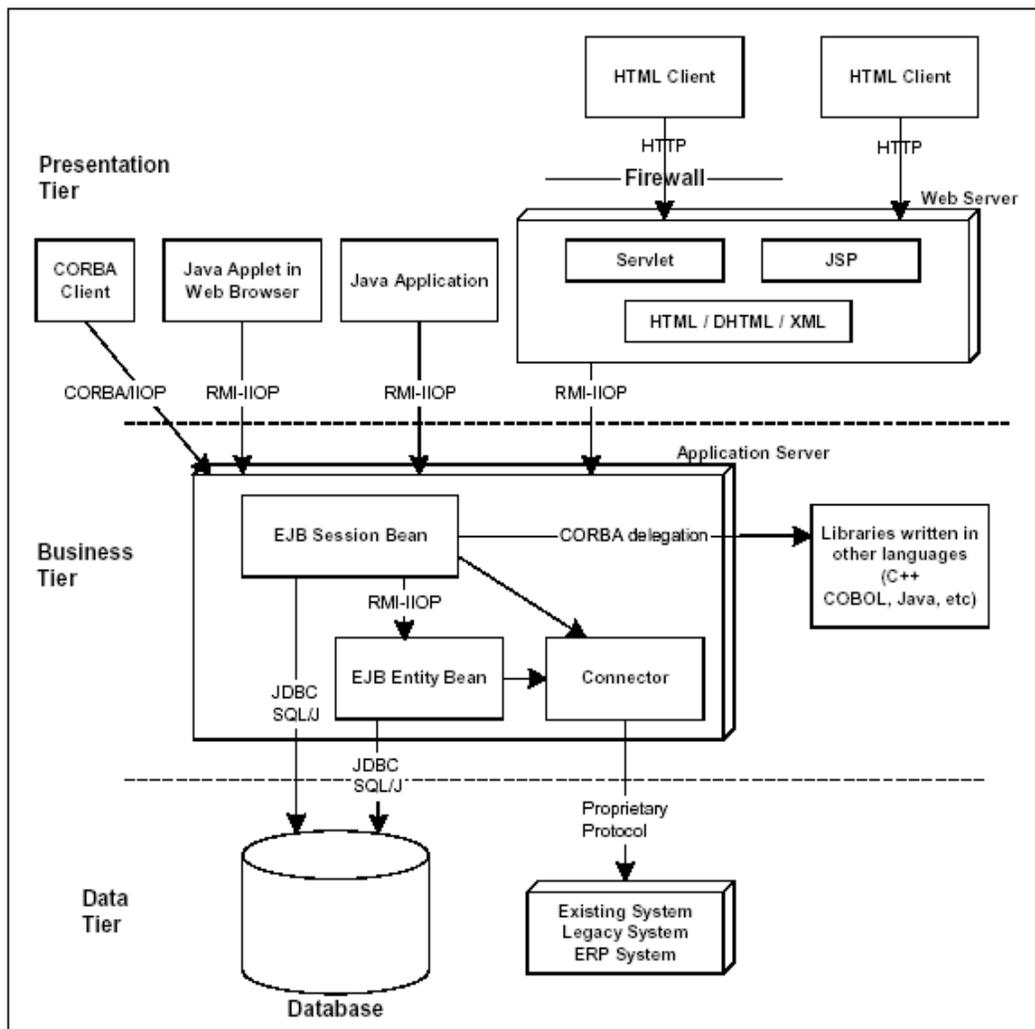


Figure 1 The J2EE Object Model

Figure 1 depicts a typical three-tier J2EE deployment. The figure is a break-down of each tier.

In J2EE, the presentation tier can include CORBA clients (clients written in languages other than the Java language), Java applets, Java applications, Java servlets, JavaServer Pages(JSP), and static Web pages.

The J2EE business tier contains business and data logic. The Enterprise JavaBeans (EJB) architecture is the server-side component model for encapsulating such logic. When a client invokes a method on a component, the J2EE platform-based product intercepts the call, and delegates it to the

component. At this point of interception, the J2EE platform-based product can perform a variety of middleware tasks, such as transactions, state management, security, or persistence. To integrate with non-Java code, one can either wrap that code in an EJB component that delegates to it via the Java Native Interface facility or wrap that code in a CORBA shell, and delegate to it via Java Interface Definition Language (Java IDL).

To integrate with databases, Java Database Connectivity (JDBC) can be used, or SQL/J can be used to interlace Java code with SQL. Integration with existing systems is performed via proprietary means, and will be standardized in the future via

J2EE connectors, which are planned to become part of a future J2EE platform release.

2.2 COM+: Windows DNA Object Model

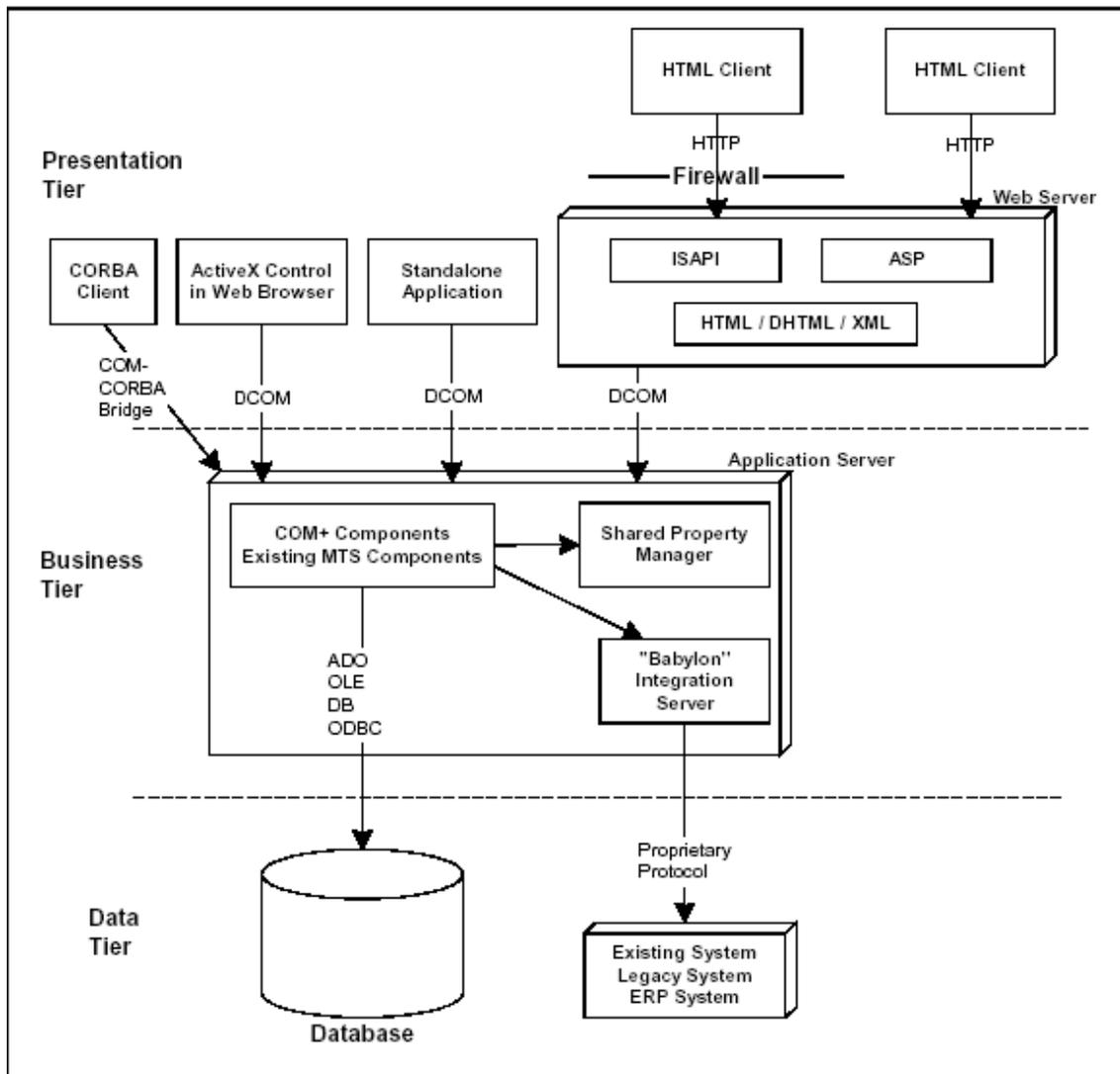


Figure 2 The Windows DNA Object Model

In Windows DNA, the presentation tier can include CORBA clients (connecting through a COM-CORBA bridge), ActiveX Controls running within a Web browser, standalone applications, Internet Server API (ISAPI) programs, Active Server Pages (ASP), and static Web pages. Clients use Microsoft's Active Directory to locate middle tier components, and use DCOM to invoke methods on those components. Although it is not shown in Figure 2 (for clarity), messages can also be sent asynchronously using Microsoft Message Queue

(MSMQ), COM+ events, or queued component technology.

The Windows DNA business tier contains business and data logic, encapsulated within COM+ components. COM+ components can be written in any language that supports COM+. All invocations to COM+ components are intercepted by the COM+ runtime, and delegated to the components. This gives the COM+ runtime the opportunity to perform

middleware operations, such as transactions, security, and object lifecycle management.

To integrate with databases in Windows DNA, one uses Microsoft's Active Database Objects (ADO) along with OLE/DB and Open Database Connectivity (ODBC). Microsoft's "Babylon" Integration Server provides for connectivity to existing enterprise information systems.

3 SCIENCE & TECHNOLOGY ARTICLES MANAGEMENT SYSTEM

3.1 Overview

This management system is just for manage all the articles of a certain school. It must initially collect the information of articles from the authors and then announce some kinds of classified statistical data. In

a word, this system should serve for a lot of persons. So we choose Web-based system, furthermore this system is not just based on Windows OS, so EJB (J2EE object model) is the appropriate choice.

3.2 Analysis and Design

In this project, we make use of object-oriented analysis and design(OOAD) and UML language to model. we also utilize some useful design patterns such as MVC and so on.

A. Web Presentation Pattern

In this opinion, the *Model View Controller* is the best choice. And then there are still left with two decisions, one for the controller and one for the view.

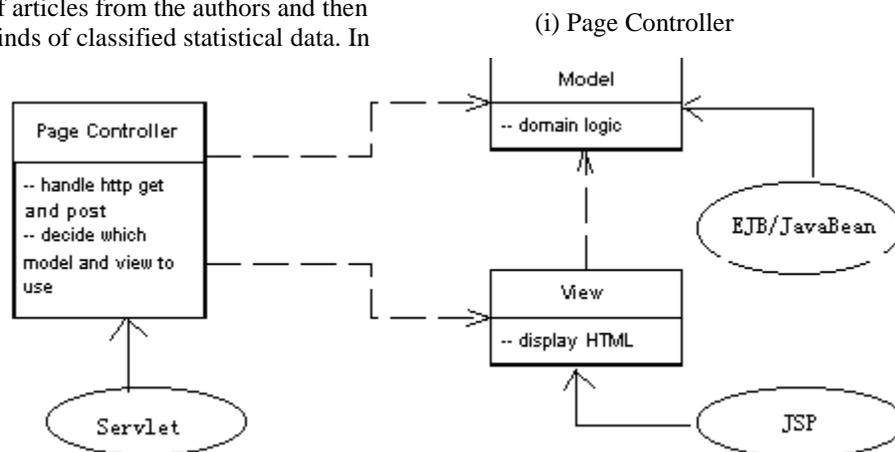


Figure 3 An object that handles a request for a specific page or action on a web site

(ii) Template View

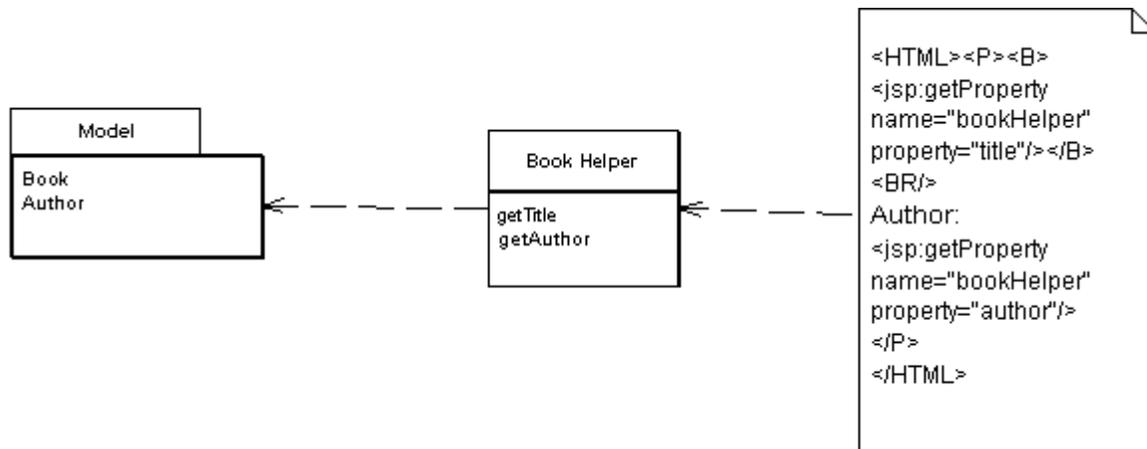


Figure 4 Render information into HTML by embedding markers into an HTML page

B. Object / Relational Mapping Pattern

As there are only several data tables in this system, and the domain logic relation between them is not very complex, so *Active Record* pattern is chosen, which is shown following.

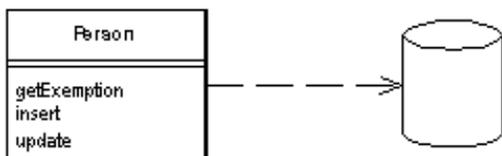


Figure 5 An object that wraps a record data structure in an external resource, such as a row in a database table, and adds some domain logic to that object

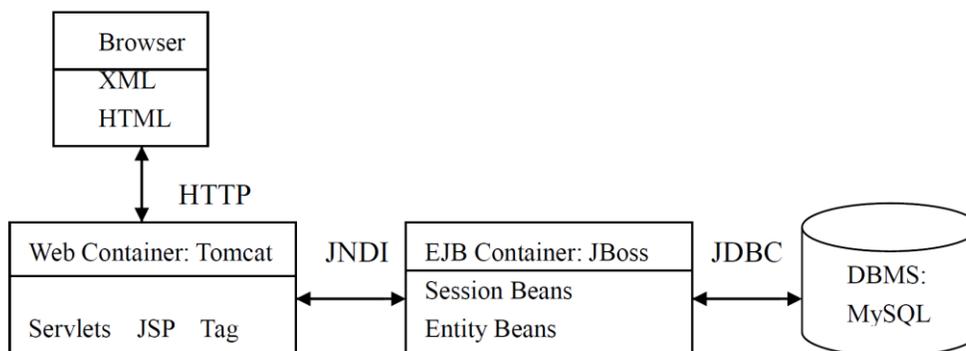


Figure 6 system architecture

4 WEB SERVICES

Note that the above technologies (EJB and COM+) can't interact each other. But When we need them interact each other, what is the solution? Fortunately, we have Web Service now.

Web Services: A collection of functions that are packaged as a single entity and published to the network for use by other programs. Web services are building blocks for creating open distributed systems, and allow companies and individuals to quickly and cheaply make their digital assets available worldwide.

Web services using XML standards is a new paradigm in the way B2B collaborations are modeled. It provides a conceptual and architectural

3.3 Implementation

We selected the following products to implement our system: DBMS: MySQL4.0, EJB Container: JBoss3.0, Web Container: Tomcat4.0, Web Server: Apache2.0, Java Develop Kit: J2SDK1.4.

As we know, all these products are *Open Source* projects software, so we can get these softwares easily. The whole system architecture is shown in Figure 6.

foundation which can be implemented using a variety of platforms and products. A web service is an application that accepts requests from other systems across the Internet or an Intranet, mediated by lightweight, vendor-neutral communications technologies. These communications technologies allow any network-enabled systems to interact.

The XML standards which a web services system is built upon allows for an implementation-neutral approach to performing business collaborations. There are many possible implementations developers can use, including a variety of products, platforms, and standards. By using a standards-based approach, developers can build a system that provides maximum interoperability for their web services.



In the solution of Web Service, there are some key web services technologies described as following.

4.1 UDDI (Universal Description, Discovery, and Integration)

UDDI is a specification for aiming towards providers and seekers of web services. The members of the UDDI Project operate a web service called the UDDI Business Registry (UBR), which is global, public directory of businesses and services. Web service providers can register and describe their services in the UBR. Users can query the UBR to discover web services and to locate information needed to interoperate with the services.

UDDI utilizes SOAP messaging (typically XML/HTTP) for publishing, editing, browsing, and searching for information in a registry. It also contains an XML schema for encapsulating the various types of data that may be returned or sent to the registry service.

4.2 WSDL (the Web Services Description Language)

WSDL specification is an XML document which describes the interface and semantics of a call to the web service. This allows for simple services to be quickly and easily described and documented.

WSDL is often mentioned along with UDDI, as the format of technical interface descriptions. While UDDI is the most common and recommended place to register a WSDL specification, the UDDI spec does not restrict what type or format of description may be linked to from its registry. It may be WSDL, a regular web page.

4.3 SOAP (the Simple Object Access Protocol)

SOAP is a specification for performing business method requests as XML documents, which offers an XML format for representing parameters and return values over HTTP. It is the communications protocol that all web services use.

XML is used because of its programming language-neutrality, extensibility, and massive industry support. HTTP is used because any Internet enabled system can communicate on a socket, because it is a simple protocol that can interoperate

with any system, and because it can navigate through firewalls using port 80, which is typically accessible.

5 CONCLUSION

In this paper, we first discuss the architecture evolution of traditional application system and Web application system; and then describe the two distributed component technologies in detail, EJB and COM+; thirdly with analysis and design the management system of Science & Technology Articles, we have carried out J2EE Object Model in detail; in the end, we describe the future trend of distributed e-business technology, Service -Oriented Web Services. As we can see, Service-Oriented Web Services will soon be the standard of distributed technology of E-Business.

Reference

- [1] Allamaraju, S. (2012). J2ee Server-side Programming. Wrox.
- [2] Fowler, M. (2009). UML Distilled II. Addison Wesley.
- [3] Fowler, M. (2008). Analysis Pattern: Reusable Object Models. Addison Wesley.
- [4] Gamma, E. (2005). Design Pattern: Elements of Reusable Object-Oriented Software. Addison Wesley.
- [5] Jie, H. (2012). The Annotated STL Source. Hust Press.
- [6] Kao, J. (2011). Developer's Guide to Building XML-based Web Services with J2EE. The Middleware Company.
- [7] Marinescu, F. (2012). EJB Design Pattern. John Wiley Composition Services.
- [8] Roman, E. (2012). Mastering EJB II. John Wiley & Sons.
- [9] Roman, E. (2009). The Business Benefits of EJB and J2EE Technologies over COM+ and Windows DNA. The Middleware Company.
- [10] Shi, J., Zhang, Y.J., Tang, X.Y., Shi, J. (2016). Modeling a Web-Based Bearing Fault Diagnosis System Using UML and Component Technology. Proceedings of 1st International Conference on Communications and Information Processing, Aveiro, Portugal.
- [11] Vawter, C. & Roman, E. (2011). J2EE vs. Microsoft.NET A comparison of building XML-based web services. The Middleware Company.



[12] Veeraraghavan, S., Panetta, K., Agaian, S. (2015). Web based Integrated Framework for Security Applications. IEEE International Conference on Systems Man and Cybernetics Conference Proceedings, Istanbul, Turkey.

[13] Wei, L. (2012). Delphi 6/Kylix 2 Soap/Web Service. China Machine Press.

[14] Wei, L. (2010). Delphi 5.x Multiple Distributed Technology of E-Business. China Machine Press.

[15] Zheng, C.Y., He, Y., Pang, X.Y. (2014). Study on the Component-based Applications System Framework. 2009 International Forum on Information Technology and Applications, vol 1, Proceedings, 684-689.