J. Software Engineering & Applications, **2009, 2:40-43**
Published Online April 2009 in SciRes (www.SciRP.org/journal/jsea)

Three-Tier Knowledge Management System Based on .NET

Mingxing Cai[1], Jintao Zheng[2], Ping Shi[1], Xiaohui Li[1]

[1]Facility Division, China University of Geosciences, Wuhan, China, [2]School of Software Engineering of Huazhong University of Sciences and Technology, Wuhan, China
Email: zheng.j.t@163.com

Received January 19th, 2009; revised February 20th, 2009; accepted March 24th, 2009.

ABSTRACT

Three-tier knowledge management system based on .NET architecture is designed according to requirement specification, characteristics of and relationship between enterprise electronic archives and knowledge management. This system using three-tier design based on factory pattern has good encapsulation and portability, with clearer and more concise structure. It degrades the costs of system development and maintenance and upgrades system's high reusability and development efficiency.

Keywords: *Knowledge Management, .NET, Three-Tier Architecture, Design Based on Factory Pattern*

1. Introduction

With the advent of the era of knowledge economy and increasing competition among enterprises, more and more enterprises are aware of the importance of the knowledge management to their development [1]. How to refine business knowledge in complex external and internal information, to effectively manage the acquisition, production and proliferation process of all kinds of enterprise knowledge, and to enhance knowledge-based capabilities, is the key to business survival. Broadly speaking, knowledge management is a positive approach to manage and optimize the enterprise's knowledge resources [2]. The goal of knowledge management is to pass on the most appropriate knowledge to the most appropriate person at the most appropriate time. It can give managers great support to make the best decisions [3]. At present, domestic researches on the enterprise knowledge management focus more on the theoretics than the practice [4]. Therefore, it is of great practical significance to design a set of easy-to-use and easy-to-promote knowledge management system, making use of the existing and somewhat general information technology, for promoting knowledge management in its application and development for the enterprise.

This paper researches and analyzes knowledge management in-depth theoretically and practically, using Microsoft .NET technology platform, to design and develop knowledge management system solutions suitable for enterprise applications, in the hope of promoting the widespread and in-depth use of knowledge management in enterprise applications.

2. Design for Three-Tier Knowledge Management System Based on .NET

The architecture design for this system is shown in Figure 1.

2.1 System Architecture Design

This system uses .NET platform based B/S three-tier architecture. B/S structure can lower the configuration requirements for the client. Three-tier architecture is a new method of programming, divided into the presentation tier, the business logic tier and the data access tier [5]. An advantage of three-tier architecture is its separation of data and format, with which programs can be easily expanded and modified to improve the coding reusability, once the business requirements are changed.

The presentation tier, using ASP.NET dynamic Web pages/sites techniques, provides users with the interoperability of the Web access interface, improving the precompiled operating mechanism, enhancing the security and guaranteeing good performance for the system.

The business logic tier contains almost all the processing functions of system business logic, which is a bridge between the data access tier and the presentation tier. Thus, the robustness, flexibility, reusability, scalability and maintainability of the software system, to a large extent, depend on the design of business logic tier [5]. In this paper, the business logic tier based on .NET platform encapsulates components such as the approval flow, business entities, statistical analysis, etc.

The data access tier offers centralized visits to the database to ensure good encapsulation and maintainability [6]. The business logic tier interacts with the database through the data access tier to avoid directly relying on the database. The data access tier, synthesizing the factory

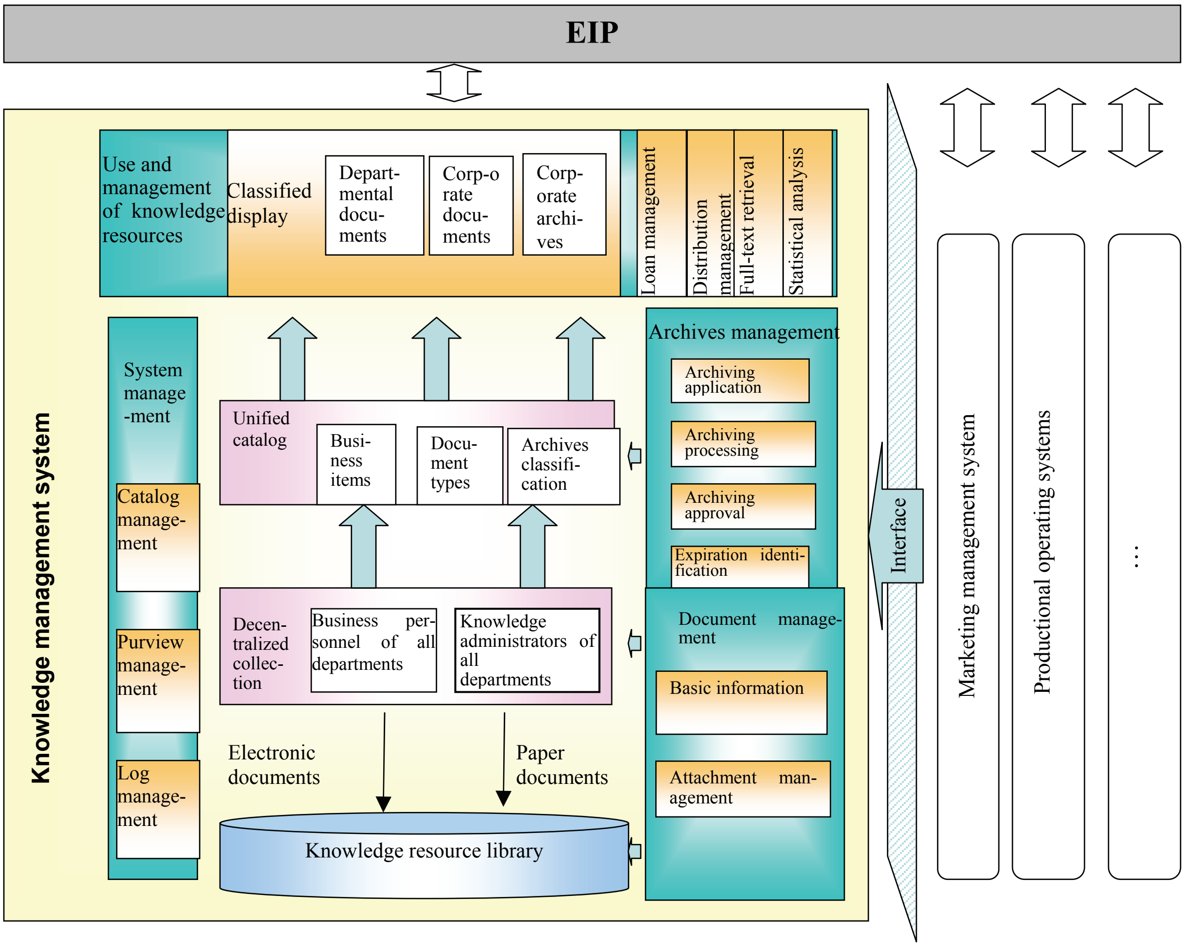

Figure 1. Architecture of knowledge management system

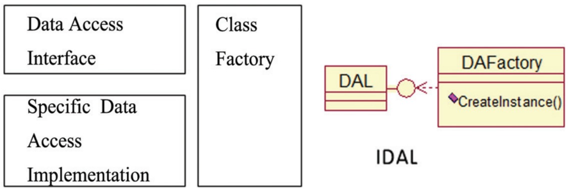

Figure 2. Data access tier based on factory pattern

design pattern and the reflex mechanism on .NET platform, makes itself more flexible, as shown in Figure 2. It really fulfills the characteristic that the data tier is unrelated to the upper tiers. That is, whatever data storage is using, MS SQL Server database, Oracle or XML documents, as long as the data is accessed through a certain interface, there is no need to revise other upper tiers and to recompile the whole system.

2.2 System Function Design

Overall flow of three-tier knowledge management system based on .NET is as shown in Figure 3. Four main functional modules are briefly illustrated as follows.

1) Document management: It mainly includes two functions of the basic information management and appended documents management, which requires business personnel of all departments to upload internally- generated documents into the knowledge management system in time.

2) Archives management: It includes four sub-modules of archiving management, expiration identification, departmental archives management and corporate archives management.

- Archiving management includes three functions of archiving application, archiving processing and archiving approval.
- Expiration identification includes three functions of expiration reminding, documents destruction and documents postponement.
- Departmental archives management module is to display, using views, the archiving application information to be processed by the departmental knowledge administrators and the archives information submitted to the corporate knowledge administrators for approval, including two schedule views of above-mentioned information. Departmental administrators have access to viewing.
- Corporate archives management is to display, mainly using views, the archives information required to be approved by the corporate knowledge administrators and the archives information already approved by them. Corporate administrators have access to viewing.

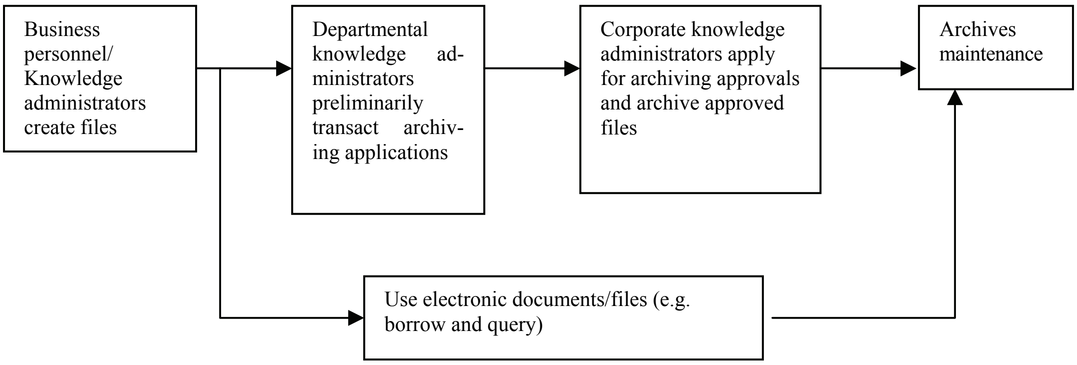
Business personnel/ Knowledge administrators create files
Departmental knowledge administrators preliminarily transact archiving applications
Corporate knowledge administrators apply for archiving approvals and archive approved files
Archives maintenance
Use electronic documents/files (e.g. borrow and query)

Figure 3. Overall flow of knowledge management system

3) Full-text retrieval: It mainly provides two functions of fuzzy retrieval and advanced retrieval. For both search modes, we have to consider the searchers' access to files. Files, to which searchers have no access, should not be included in the search results. Advanced retrieval function is to rapidly integrate related information scattered in various heterogeneous systems and then present it to users, according to the meanings of domain knowledge possessed by the information resources. The implementation of this search function is based on Conceptual Set C and Attribute Set A in the ontology =(N,F,C,A,R,R), accurate relationship between them and comprehensive setup of the ontology [7].

4) System management: It is mainly composed of five sub-modules such as classified catalog management, file storage settings, access management, log management, and full-text retrieval management.

3. System Implementation

As for three-tier architecture based system, there are three main assemblies-KMS. Web. dll for the presentation tier, KMS. BLL. dll for the business logic tier, and KMS. DAL. dll for the data access tier. And else there are three assisted assemblies-entity class KMS. Model. dll, public class KMS. Utility. dll, and class factory KMS. DAFactory. dll [8].

In order to implement real separation of data tier and other upper tiers, class factory design pattern and reflex mechanism design are integrated to fulfill the data access tier [8]. Using factory pattern + reflex mechanism + caching mechanism, dynamic creating different object interfaces to data tier is implemented.

1) Define an interface and a basic method used to access users' information [9].

```
namespace KMS.IDAL
{
    public interface IKMAdmin
    {
      /// add new data
      void Add(Maticsoft.Model.KMAdmin model);
    }
}
```

IDAL interface project is only a statement of a series of methods, while specific implementation details will be accomplished by data access project.

2) Compile corresponding data access project for Sql database

```
namespace KMS.SQLServerDAL
{
    public class KMAdmin : IKMAdmin
    {
      public KMAdmin()
      { }
      /// add new data
      public void Add(KMS.Model.KMAdmin model)
      {
          // ellipsis
      }
    }
}
```

Data access project SQLServerDAL fulfills "add", "modify" and "delete" functions for records in connection with the Sql database.

3) Create a class factory "DALFactory" to identify which database users choose when they are building applications.

DALFactory read the database identification information from the system configuration file "web.config" to provide users with a unified interface, making users worry less about it.

When the Sql database is used, information identifiers in configuration files are as follows:

```
<add key = "DAL" value = "KMS.SQLServerDAL">
```

DALFactory project will select the corresponding database according to the database identification information.

```
  private static readonly string AssemblyPath = ConfigurationManager.AppSettings["DAL"];
        /// <summary>
        /// create an object or obtain from cache memory
        /// </summary>
        public static object CreateObject(string AssemblyPath, string ClassNamespace)
        {
             object objType = DataCache.GetCache(ClassNamespace);// read from cache memory
             if (objType == null)
             {
                  try
                  {
                       objType = Assembly.Load(AssemblyPath).CreateInstance(ClassNamespace);// generate reflection
                       DataCache.SetCache(ClassNamespace, objType);// write cache memory
                  }
                  catch
                  { }
              }
              return objType;
        }
        /// <summary>
        /// create data access interface of KMAdmin
        /// </summary>
        public static KMS.IDAL.IKMAdmin CreateKMAdmin()
        {
             string ClassNamespace = AssemblyPath + ". KMAdmin";
             object objType = CreateObject(AssemblyPath, ClassNamespace);
             return (Maticsoft.IDAL.IKMAdmin)objType;
 }
```

The system encapsulates the processing of the lower tiers into the project and provides interfaces to the logic tier, so that it can be directly called.

4) Examples for calling interface of the logic tier.

```
namespace KMS.BLL
  {
   public class KMAdmin
      {
       private readonly IKMAdmin dal = DataAccess.CreateKMAdmin();
       public KMAdmin()
           { }
           /// add new data
           public void Add(KMS.Model.KMAdmin model)
           {
           dal.Add(model);
           }
      }
   }
```

In this way, the encapsulation and universality are fulfilled well.

4. Conclusions

Three-tier knowledge management system using .NET architecture design has good reusability, encapsulation and scalability. Compiled programs are of high security and efficiency. Separation of interface and programs makes the structure clear, which is easy to maintain. Through the design for every module in the knowledge management system, giving support to current archives management work is fulfilled. It combines archives management with knowledge management to promote internal knowledge sharing.

REFERENCES

[1] B. Gallupe, “Knowledge management systems: Surveying the land-scape [J],” International Journal of Management Reviews, 3(1), pp. 61−77, 2002.

[2] J. X. Zhou and R. R. Zhou, “Research on PLM system framework and key technologies,” Transactions of Nanjing University of Aeronautics & Astronautics, Vol. 35, No. 5, 2003.

[3] R. Mack, Y. Ravin, and R. J. Byrd, “Knowledge portals and the emerging digital knowledge work place,” IBM Systems Journal, Vol. 40, No. 4, 2001.

[4] C. E. Wang, Y. P. He, and Q. L. Shu, “Product life cycle modeling and management,” Science Press, Beijing, 2004.

[5] S. S. Wei and N. S. Xu, “Design and implementation for data monitoring client based on .NET components,” Computer & Digital Engineering, No. 6, 2008.

[6] Q. Z. Wu, “Operations research and optimization method [M],” China Machine Press, Beijing, No. 8, pp. 218, 2003.

[7] X. L. Liu, H. R. Wang, and G. J. Yang, “Research and development of ontology-based knowledge management system,” Journal of Hebei University, No. 5, 2008.

[8] S. Li and J. X. Wang, “Design and implementation of enterprise manager assessment & evaluation system based on .NET three-tier B/S architecture,” Journal of Nanjing University, No. 3, 2007.

[9] K. Zeng, “Design and implementation of B2C mobile e-commerce website based on ASP. NET,” Computer & Digital Engineering, No. 3, 2008.