A Dynamic Framework for e-Commerce Portals

Man Hing Yu, Chee Chern Lim, Jesse S Jin

School of Design, Communication and I.T., University of Newcastle, NSW 2308, AUSTRALIA

Summary
The objective of this paper is to enhance advanced e-commerce framework by supporting a complete customization of the data structure and layout presentation via portals technologies. This paper introduces a scheme using the generic data model in the database structure and interface-design toolkits, which allow the users to make modifications on the data attributes while the application has the ability to adapt these changes in the templates and layout presentation. The approach achieves greater adaptability, extensibility and reusability of database structure in the e-commerce applications. In addition, it discusses the integration of Content Management System and Business-to-Business web services compositions. This is achieved by three main portals: Business-to-Consumer portals, Content Management System (CMS) portals and Business-to-Business portals (optional/additional services).

Key words:
Dynamic framework, adaptive e-commerce, portal.

1. Introduction

Today, the business community has realized the portal solutions as an opportunity to develop and maintain integrated and personalized environments for e-commerce. Based on the natural behavior of individual portal, portals have been classified into Business-to-Consumer portals, Content Management System portals and Business-to-Business portals in this paper. While the role of individual portal is different from each other, however, the ultimate objective is to deliver adaptive e-commerce and manageable e-commerce solutions via portal technologies.

Enterprise Resources Planning (ERP) is a cross-functional enterprise system that serves as a framework to integrate and automate business processes such as manufacturing, distribution, accounting, finance, logistics and human resources. ERP provides significant efficiency and improvement in company's business processes. The leading vendors for ERP software are SAP, Baan, J. D. Edwards, Oracle and PeopleSoft. The most popular product is SAP R/3, developed by a German firm and its newer web-based variant mySAP.com, which allows its users to work via the World Wide Web [1, 2]. SAP R/3 is a client/server system employing a common, integrated database with shared application modules; it handles both TCP/IP and SNA communication protocols [3, 4]. Many ERP vendors are moving existing desktop solution to web portal solution due to the rapid advancements in Internet technology and telecommunications. Products like SAP consist of numerous modules; which provide flexibility and variety to make changes to the whole system. Nonetheless, the modification of the data structure involves the interaction of technical developers, which are expensive and time-consuming. It is mainly due to the user acquire new capabilities with growing familiarity of the system [5] and frequent changes in business such as new product development [6].

This paper introduces an e-commerce portal that utilizes the generic multi-media ERP (GMM ERP) architecture. There are several significant differences between GMM ERP and traditional transaction systems. First, the traditional system is integrated through a common set of definitions and pre-defined schema in database. Second, modules are tightly coupled and integrated in the system. The main advantage of the GMM ERP provides a complete customization that supports wide-range of business needs. The major advantages of the proposed e-commerce portal are the adaptability and extensibility of the data structure. By utilizing the approach of generic database, the e-commerce portal allows complete customization of the database schema. For instance, the current approach is to create a predefined data structure/schema before the database can be used for storage. In this traditional approach, the application is restricted to operate in a confined static or fixed data structure environment, and the user is not able to store other additional non-predefined attributes information. However, the generic multi-media ERP can be adapted to support significant changes in trading relationships and alteration of data structure. In addition, the e-commerce authoring portal is user-friendly, efficient and cost effective.

This article starts with the introduction of the generic multi-media ERP architecture that achieves adaptability, extensibility and reusability in the system. Next, it provides detail information on the database connection layers that support the generic multi-media data model. Finally, the paper concludes with the three main portals: Business-to-Consumer (B2C) portals, Content Management System (CMS) portals and Business-to-Business (B2B) portals.

2. Dynamic Framework

2.1 Data Models of the Interface

In the proposed framework, there are two kinds of data input and output model being used in the: Product Data Management (PDM), and Workflow Management System (WfMS) [7]. These two data models can provide the best performance in information exchange and workflow coordination in the proposed system. The PDM is used to control access to documented versions of product designs,

Manuscript received February 26, 2006.
Manuscript revised March 24, 2006.

which include the traditional single data record such as product details and company contact details. It plays an important role in the storage and access of data and documents throughout the process. Conversely WfMS allows managers to coordinate and schedule the activities of business processes to optimize the flow of information between partners and resources. It is used to coordinate the more complex and repeatable work processes of production. Based on these two data models, the system will prepare a data input and retrieval interface for the access of logistic information online.

2.1.1 Product Data Management (PDM)

The PDM is an ideal model to control product and contact information, which involve two kinds of database structure design: one entry of table storage and multiple entries of table storage. The one entry of table storage satisfies the table schema which has a one-to-one relation with other tables, and the purpose of this table is only used for storing and accessing records. For example, contact management of supply chain members, will normally only have one single table called 'member' to store and access the record. However, the multiple entries of table storage, when one table has been normalized by 3-N rules, results in splitting into a one-to-many relationship with another table. For example, one invoice will have multiple purchase item details, which establish a one-to-many relationship within the entry. Hence sales management is classified by this kind of data input.

2.1.2 Workflow Management System (WfMS)

WfMS category is concerned with time-related tasks, which can handle any re-structuring, such as roles and responsibilities of parties in the supply chain logistic route. The system can automatically detect time conflicts in the existing route. When the administrator arranges for a workflow [8], which may require member A, the system will check for the availability of member A, and wait for guarantee from member A about the delivery in real time. If resource conflicts are identified the system will inform the user about the next best option.

2.2 Three-Layered Generic Database Architecture

The objective of this paper is to provide the flexibility of data medium and adaptability of the presentation interface. Today's business system requires that the database structures are adaptable, extensible and reusable, to enhance the value of the business. The three-layered generic database structure in the proposed system design structure is the solution for this. The physical database layer is the lowest layer in the architecture; followed by the database structure layer and finally the database connection layer (see Figure 1). The physical layer is responsible for handling traditional storage medium such as RDBM database and XML [9] files.

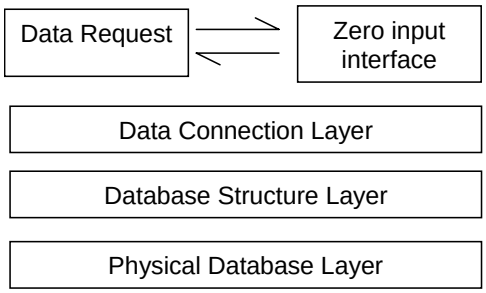

Fig 1. Three-Layered Architecture

The middle layer, “database structure layer”, contains objects that are responsible for managing the table schema in the database. All high level queries are passed and organized by the run-time objects in this layer. These queries are broken down into simple standard SQL access or storage operation and passed to the physical layer for execution. The upper layer, “database connection layer” maintains the connection between the system program and the database. This architecture allows the replacement of alternative storage medium in the physical layer with minimum effort. In addition, it allows flexibility and instant modification of the database scheme structure. This design breaks through the traditional fixed or static database schema and achieves the aim of flexibility in database structure. The three different layers will be explored separately in this section.

2.2.1 Data Connection Layer

This layer provides the interface allowing the program accessing the database system. The DB (*$szHost*, *$szDatabase*, *$szUser*, *$szPassword*) is the constructor of the interface; it connects to the storage by providing the unique identification. The user can request the service by the query (*$szQuery*, *$QID*=0). The *next_record* (*$QID*=0) function can be used to retrieve the query result with the Query ID (*$QID*). In addition, this function performs errors checking, prepares the current result row in the memory and increases the row counter for operation efficiency. The *current_record* (*$QID*=0) returns the current array of record in the memory. The result will be released from memory once the free (*$QID*=0) is executed.

2.2.2 Database Structure Layer

The database structure layer is the most important layer because its task is to organize the data structure in the database. Each original table structure is represented by *myTable* and *myField* (see Tables 1 and 2). Both tables are identified by a unique ID: *TID* and *FID* respectively. The *TID* in *myField* table is a foreign key from *myTable*, which allows one table to contain multiple fields. The structure of these tables is used to represent the core information in database schema. The combination of these

tables achieves very high flexibility in the database structure. Each data structure is clearly defined, and allows changes in the *myField* table. These two tables are allocated in a separate database due to their frequent high access rates. Lastly, the *myValue* table (see Table 3) contains the actual data storage of all the data medium contents. All these values are referenced by the unique primary keys *TID* and *FID*, which have been defined in the *myTable* and *myField* tables.

Table 1. myTable structure

TID	Auto increment int
TableName	Text

Table 2. myField structure

FID	Auto increment int
FieldName	Text
FieldType	Text

Table 3. myValue structure

TID	Int
FID	Int
Value	Blob

For example, the system uses a traditional schema table "Student" to store the student information (see Table 4). It will store the student number, student name and the age of the student. The database structure layer will represent the data structure in *myTable* and *myField* tables based on the generic presentation approach. Tables 5 and 6 show the *myTable* and *myField*, which contain the original structure of the table "Student". The *TID* and *FID* act as the key to access the values stored in the *myValue* table, see Table 7. This design separation resolves the dependency of data values on the data structure, and achieves a greater flexibility of dynamic data structure in the database system.

Table 4. Original student table values

StudentNo	Name	Age
1001	Alex	20
1002	Bob	25
1003	Cathy	30

Table 5. myTable value

TID	TableName
1	Student

Table 6. myField value

StudentNo	FieldName	Type
1	StudentNo	Text
2	Name	Text
3	Age	Int

Table 7. myValue value

TID	FID	Value
1	1	1001
1	2	Alex
1	3	20
1	1	1002
1	2	Bob
1	3	25
1	1	1003
1	2	Cathy
1	3	30

From traditional web system development, changes in the data structure will likely involve tedious work and maximum time in order to make the necessary modification from the old data structure to the new data structure. It involves changes in the database schema and the actual codes in every database-driven web page. This might actually involve many complications for just a very small change in the schema. For example, if the previous "Student" data structure requires an extra attribute "address", most developers will have to manually change the schema in database and modify every page containing the "Student" entity. However, with the approach of the dynamic framework, developers are only required to make changes in the *myField* table through the user-friendly web authoring interface without any further complications [10]. This design approach also overcomes the problem in contents changing due to client uncertainty at the commencement of research, and eventually provides a flexible and dynamic data structure in the web prototyping development process.

2.2.3 Physical Database Layer

This layer acts as an adaptor interface between the database structure layer and the physical storage medium. The existence of this layer allows the connectivity with different storage medium. Any changes of the medium only require the modification of this layer functions, such as the connection establishment command, and the query interface of the new medium. Hence, it leaves the database connection layer and database structure layer unchanged, increasing the compatibility and extensibility of the whole model.

2.3 Abstract Program Logic Interface

The design method adopted in this research was a layered approach where each module is represented as layers in the system (see Figure 2). As a general rule for good

programming, it is suggested that components such as presentation and authentication be separated from the business logic [11, 12]. It is for this reason that the layered approach was employed and it offered an effective way for independently coding the various components without any affects to the functionality of the system. It also ensures efficient and easy connection between the database server and the application programs. The design of the individual module applies the concepts from object oriented design techniques and general coding methods and style, because each module possessed different prosperities and thus it required different design techniques and coding styles. In the following section, this paper will focus on the user interface module and the dynamic form generator that achieve the aim of dynamic framework.

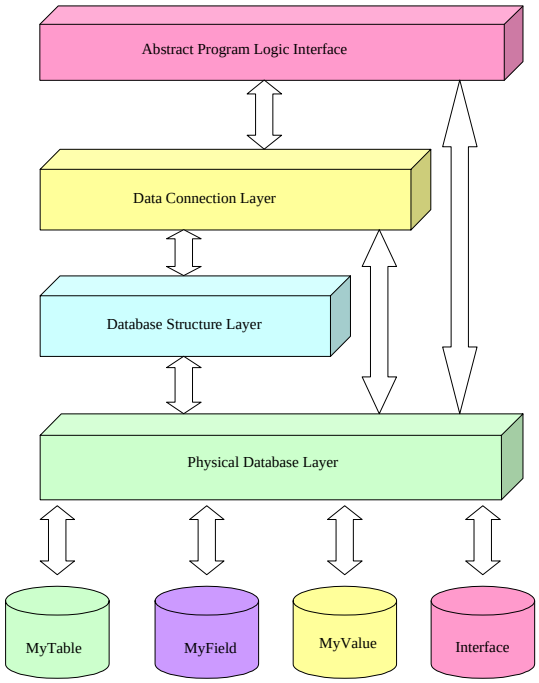

Fig 2. Framework Architecture

The user interface module provides an interface for the user to interact with the system, and also provide the user-friendly compatibility [13]. Its main function is to gather user input and pass the input command with data to the System Logic and Data Validation modules to process the user's request. Likewise, this module will interpret the result generated from the System Logic module and display the result back to the user interface. The User Interface module was designed with web page modification and creation simplicities in mind. This module was developed in a way that removed the need for programmers to hard code each individual web page. The objectives include:

- To design a style sheet to ensure consistency between web pages.
- To create an interface for the creation, modification and deletion of HTML forms needed for business transaction.
- To provide an easy mechanism for the generation of menu bars required for the website

The drive behind this module was the need for an interface for the user to interact with the system. Due to the problem of usability and consistent screen design, a system was required to manage and create the screen design for each page in a consistent and usable manner. In addition, an efficient mean for the dynamic generation of these form pages is essential to reduce the need for hard coding every page individually because of the large number of form pages required for each area of the business. The User Interface module was designed with web page modification ability and creation simplicities in mind. This module was developed in a way that removed the need for programmers to hard code individual web page. In addition, the User Interface modules are tightly coupled with the whole system. All modules that generate a web page must take advantage of the Cascading Style Sheet to define their pages. The main sub-modules include the various components that are used to develop the interface for users:

- *Cascading Style Sheet*: All formats of web documents are defined in the CSS. This is used to provide consistency in page appearance across the whole system.
- *Dynamic Form Generator*: Allows users to build HTML form dynamically through functionalities to add, edit and delete form elements. Querying the database and transforming the results into a HTML form through the use of web scripting language to achieve this.
- *Dynamic Menu Generator*: Allows users to add, edit and delete links on the side menu bar and the content page menu bar. This interface module builds the side menu (main menu) and the content menu (sub menu).

Next, the paper will introduce the three main portals: Business-to-Consumer portals, Content Management System portals and Business-to-Business portals. These portals are developed by utilizing the dynamic framework to achieve greater adaptability, extensibility and reusability in the e-commerce applications.

3. Integrated Portals

The business community has envisioned portal solutions as an opportunity to develop and maintain integrated, personalized environments for e-commerce. Based on the natural behavior of individual portal, portals have been defined as Business-to-Consumer portals, Content Management System portals and Business-to-Business portals in this article.

3.1 Business-to-Consumer Portal

A Business-to-Consumer portal consists of the following portlets:

- Products browsing
- Information pages browsing
- Shopping cart portlet
- Tell a friend
- News subscription
- Mailing List
- Negotiation Agent

The attributes information of products is alterable with the unique generic data structure based on the dynamic framework. The information pages are creatable and editable via the WYSIWYG editor which it provides great benefits for users without skills of web programming. These portlets are managed via the Content Management System portals.

3.2 Content Management System Portal

A Content Management System portal includes the following modules:

- Category management
- Product management
- Web contents / pages presentation management
- Customers management
- Subscription management
- Mailing List Campaign
- Order / Transaction management

This portal provides the user a full control over the content, description and cosmetic appearance of the online store.

3.3 Business-to-Business Portal

The Business-to-Business portal is an additional portal and can be activated if a B2B relationship exists in the company. The B2B portal provides an advanced infrastructure and complicated functionality that supports a range of B2B activities such as:

- Wholesale customers management
- Supplier management
- Wholesale customer online login management
- Shipping / Purchase order / Order receive management
- Stock inventory / product faulty management
- Reserve/ Invoicing System
- Payment transaction tracking
- Credit Notes management
- Login account management
- Retails Shop account management
- Customised Reports such as Profit and Loss, Outstanding Delivery.
- Calendar and Communication module

These modules are integrated in the B2B portal and designed with supply chain concept and workflow.

4. Conclusion

In a society that not only expects, but rather demands high quality in all areas of Information Technology, the e-commerce domain is experiencing an extraordinary influx of new process and design innovations. From the moment a prospective client enters an e-commerce site to the moment they exit, after purchasing or not, sites are becoming smarter by analyzing the movement and selections made by consumers to better serve the next client entering the store. Previously, however, these high-end, dynamic and adaptive sites were only available to the large corporation willing to inject the required resources, such as time, money and manpower, to develop such an application. The lack of a simple, yet extremely effective e-commerce application builder in the marketplace, for the SME and MNE alike, has led to the development of the e-commerce portals.

In this paper, the e-commerce portals are introduced with its unique approach to integrate the B2C and B2B into one e-Commerce portal and provide advanced CMS to configure and manage the portlets. By utilizing dynamic framework in the data structure presentation, it achieves the adaptability, extensibility and reusability of database schema in the e-commerce applications. Last, the web portal with the centralize system management, resulting in more efficient and less complicated data management.

References

[1] M. Perez; T. Hantush; B. Matzke, The Sap R/3 System on the Internet. Addison-Wesley, Reading, MA, 1999.

[2] C. Larsen, Fusing of e-Business and ERP. SAP America, 2000.

[3] H. L. Lee & S. Whang, "E-Business and Supply Chain Integration", Stanford Global Supply Chain Management Forum (SGSCMF-W2-2001), November 2001.

[4] H. L. Lee & C. Martin, "Supply Chain Confidence", Global Trade Management, Vastera, Inc. 2001.

[5] N. L. Junco and R. A. Bailie, "A case study of content management," presented at Professional Communication Conference, 2004.

[6] N. Junco, R. A. Bailie, and D. Ledet, "A case study of a content management system: choosing and implementing a CMS," presented at Professional Communication Conference, 2005.

[7] P. Kearney, "Integrating A.I. planning techniques in workflow management system," Elsevier, 2002.

[8] H. L. W. Lee, S, "E-Business and Supply Chain Integration," presented at Stanford Global Supply Chain Management Forum (SGSCMF-W2-2001), 2001.

[9] L. K and B. M, "XML, metadata and efficient knowledge discovery," Elser, 2000

[10] C. C. Lim, M. H. Yu, and J. Jin, "Rapid Prototyping for Web Applications," presented at WAIM, 2004.

[11] Q. Li, "Study of A Content Oriented Web Architectural Model," presented at IEEE, 2001.

[12] P. Russell, "Infrastructure - make or break your e-business," presented at TOOLS-Pacific, 2000.

[13] S. Overmyer, "What's different about requirements engineering for web sites?," Requirements Engineerng Journal, vol. 5, pp. 62-65, 2000.