

E-COMMERCE SOLUTIONS WITH ORACLE e-DATABASE

Ivan Matasić, Ivan Magdalenić, Hrvoje Šimić

Faculty of Electrical Engineering and Computing

Dept. of Telecommunications

University of Zagreb

Unska 3, 10 000 Zagreb, Croatia

Tel: +385 (1) 6129 686 Fax: +385 (1) 6129 832 e-mail ivan.matasic@fer.hr

Tel: +385 (1) 6129 686 Fax: +385 (1) 6129 832; e-mail ivan.magdalenic@fer.hr

Tel: +385 (1) 6129 686 Fax: +385 (1) 6129 832; e-mail hrvoje.simic@fer.hr

Abstract: *Economical transition of countries such as Croatia into modern information and communication technology based communities and large-scale adoption of e-business practices is a complex task. Some businesses will certainly be able to cope with technological and business issues that are related to these practices, but the vast majority will be not. If nothing is done to assist the acceptance of new modes of operation, they will be left aside losing the battle in competitiveness with other service and goods providers. The paper describes e-mall application residing on Oracle internet database Oracle 8i that copes with these problems, creating infrastructure base that will introduce businesses into Internet economy world.*

KEYWORDS: *E-commerce, Oracle DBMS*

INTRODUCTION

A research program, called CETT e-mall and aimed at setting up e-business support services and applications, had been launched. It is intended that an operative informational business centre provide e-applications service for large, small and medium size enterprises, selling goods, informations and services via Internet. By re-allocating complexities in the system as a whole, which should be done through hosted services, companies with limited human and other resources will also be given the possibility to benefit from B2B and B2C interactions. Implementation of this application and is aimed as a model especially important for raising efficiency of Croatian companies in terms of strong international competition.

Self-service is one of the basic components of e-business support services. The CETT application is organized in a way that ensures independence from data storage. Oracle 8i enabled storing all data and business logic into the database, so nothing has to be stored in file system. System is compact, open and easily expandable. Flexibility and modularity of the system enables easy upgrades and new features incorporation. All business logic and functionality of application is withheld within database and application server. Problems with Multilanguage and MultiCurrency are solved in an efficient and unique way, as are considered to be very important in small-market countries in transition. Implementation of Multilanguage and MultiCurrency preserves the modularity of application.

The paper is organized as follows: Business logic is explained in chapter 1. Logical structure of application is described in chapter 2. Chapter 3 presents database schema and PLSQL packages stored in database and is followed by Conclusion and References.

1. BUSINESS LOGIC

Every dynamic Web site should be decomposed into four "ingredients": structure, content, look and functionality. **Structure** defines the purpose of every page and relations between the pages, thus effectively connecting the other ingredients. **Content** ingredient is responsible for all the materials in the site perceived by the user as the "content", as opposed to the look of the page. **Look** of the pages enwraps the content for a more pleasurable and effective human experience. **Functionality** captures all the specific processing involved with the site, the Web application itself.

Ingredient elements can be collected into sets, and the site itself is defined by the four such sets, one for the each ingredient. It is necessary for these sets to be compatible for the site to work properly. But also, the compatible sets of the same ingredient can be interchangeable, allowing the change in one "dimension" of the site and leaving the others intact (e.g. visual redesign).

Standard tree tier architecture was used, with database tier for complete storage of business and platform information, which includes not only standard data types but also multimedia; application tier for business logic and Web platform and thin client tier for displaying. Figure 1. presents complete e-commerce structure implemented through standard three tier architecture.

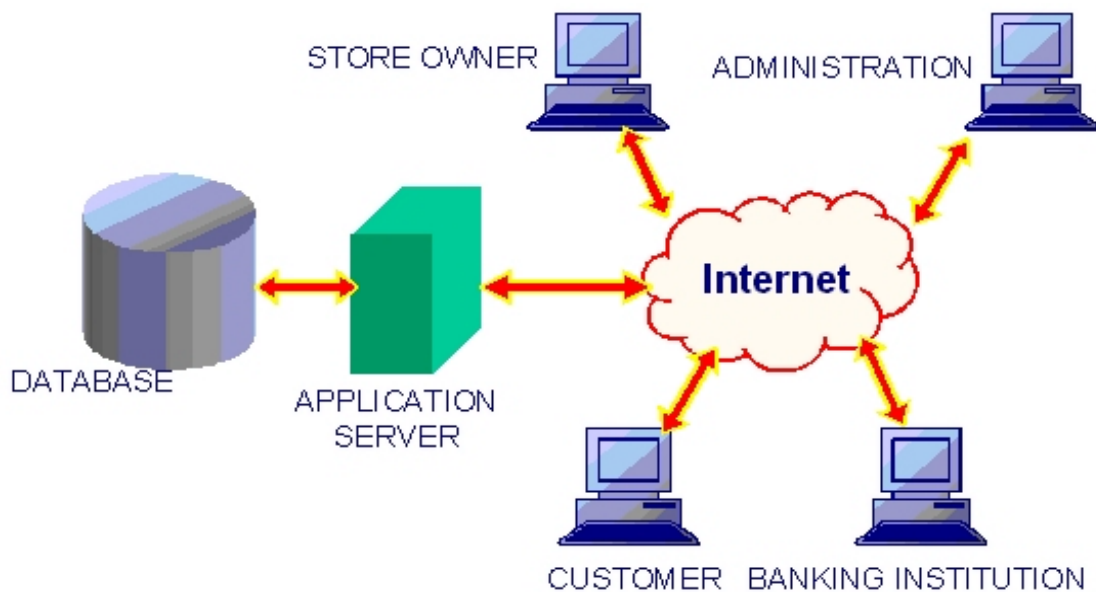


Figure 1 - *E-commerce*

The application is organized in a way that ensures independence from data storage. Oracle 8i™ object-relational database management system (ORDBMS) is used to store both data and application [1], [3].

The business logic layer can be analysed from two points of view, technical and logical. From the technical point of view, as presented in Figure 2., the business layer **structure** and **functionality** is modelled with PLSQL procedures stored in the database. These procedures implement business logic of a e-commerce site. Web templates are separated PLSQL files that define graphical presentation (**look**) of the data (**content**) (in all formats including pictures and video) from the database. Web templates define structure and look-and-feel of a html page.

Oracle 8i enabled storing all data and business logic into the database, so nothing has to be stored in file system. So, the system is more compact, open and easily expandable. The advantages in management of such data are huge. Flexibility and modularity of the system using well-defined PLSQL procedures [2] enables easy upgrades and new features incorporation into e-commerce application. All business logic and functionality of application is withheld within database and application server, so the client doesn't need to take care of any software and the data is placed in a safer way.

From the logical point of view, the business layer is divided into three sections: tourism, goods and custom designed ACI Marine applications. These will be explained in following section.

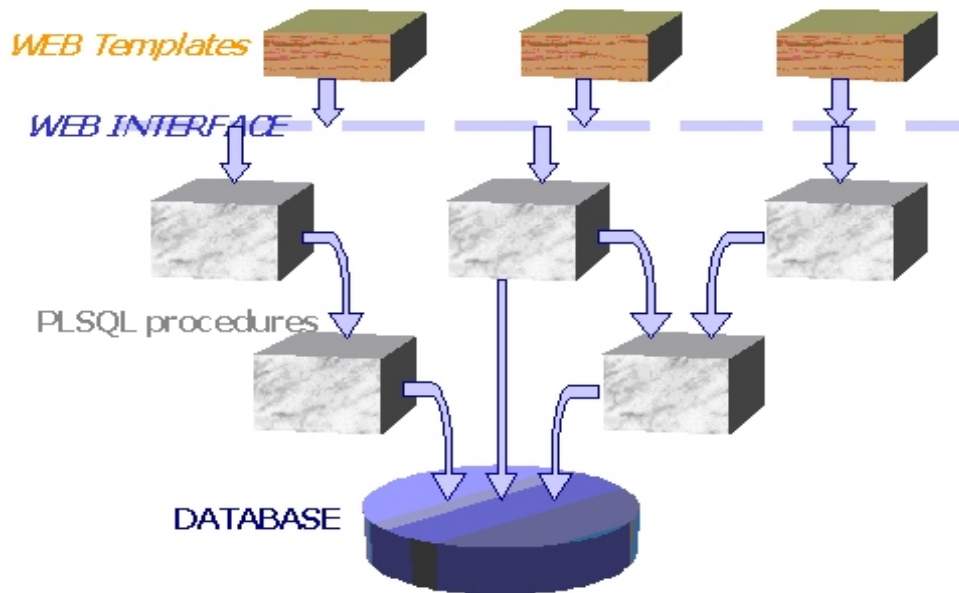


Figure 2 - E-commerce

2. LOGICAL STRUCTURE OF APPLICATION

Before we explain how application was structured logically, we will address some features that each user (customer or owner) must have (or expects) from dynamical web site. Besides security and reliable customer-merchant-bank transaction management, each customer expects searching through site as an authorised or non-authorised visitor, custom definitions of look-and-feel parameters for each authorised user (session control must be implemented), unlimited number of addresses (for delivery purpose), shopping list, wish list, basket ... and so on. Each customer would also expect reports on his orders and spending, as each store owner needs business, statistical and financial reports. The modularity of the system enables definition of custom reports for each store owner.

2.1 Logical division of the e-commerce site

As stated, from the logical point of view the business layer is divided into three sections: tourism, goods and custom designed ACI Marine applications.

2.1.1 Selling goods

This kind of e-commerce is well known from many web sites including amazon.com and similar trading dot.coms, so here we will emphasize on those features that are not widely spread through

these sites. In e-commerce with goods, application offers flexible structuring of stores and items as seen both by the owner and the customer. The store owner is able to organize his store in the most logical way. It is possible to get to the same item through many different ways, as shown in Figure 3.

The store owner can structure his store in-depth, at his own wishes and can describe each article with his own attributes. He can add links to products in his own store or some other store. Store owner can select web templates offered by application provider change their parameters (like background colour) or can define his own custom designed web templates.

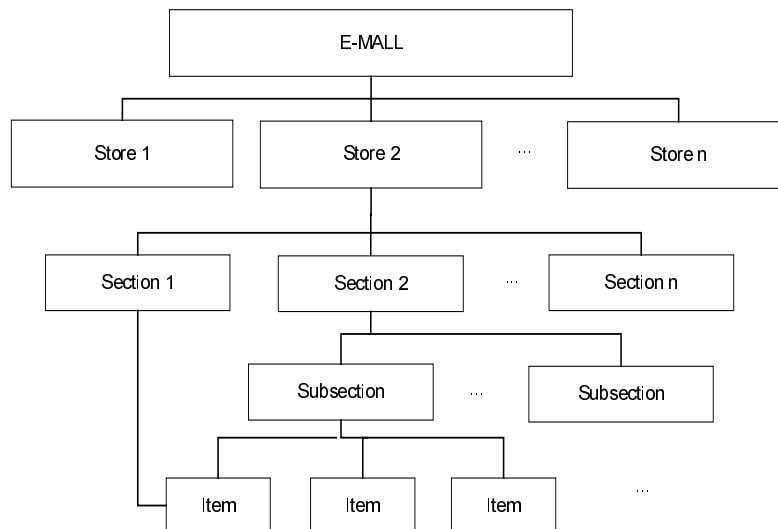


Figure 3 - Example e-mall structure

2.1.2. Tourism

Selling services such as tourist beds differ from selling goods in business logic as well as demands different data structure. Product in this case is a unique element created at the moment of order. For example, single bed bedroom with sea-view and air condition is not a product, but single bed bedroom with sea-view and air condition from 01.07.2001 until 31.07.2001 is. We further concluded that all these 'products' can be categorised with limited number of product types (single bed room with TV is one type of room). Therefore it is possible to improve e-mall functionality and enable the store owner (in this case hotel owner) more efficient and less time-consuming data entering (containing descriptions of his offering). Problem of booking the rooms, that is communication with hotel so that hotel and web application would not sell the same product twice must also be addressed properly. These and many other problems ensured that business logic for tourism should be somewhat different, while e-mall must contain its unique presentation interface toward customers.

Beside data and business logic needed for tourist services trading, we felt that complete description of this field should include related sights and events. This would enable all tourist institutions beside hotels and resorts (like tourist offices in certain region) to incorporate with (private or public) tourist facility sector in presenting town, region or other type of tourist destination and therefore attract more costumers to it. Each event or sight description is highly connected with proper multimedia data stored into database. 'Upload' of pictures connected with sights or events description was very efficiently solved through application.

2.1.3. ACI Marine custom design application

As stated previously, large-scale adoption of e-business practices is a complex task, particularly when small and medium-size enterprises are expected to play a substantial role in the development and restructuring of economies. Some of them will certainly be able to cope with technological and business issues that are related to these practices, but the vast majority will be not, and if nothing is done to assist the acceptance of new modes of operation, they will be left aside from the mainstream.

Some large enterprises will certainly have different kind of requirements when transferring to e-commerce: business logic will need to be modified for their successful web presentation and commerce. We have therefore decided to verify modularity of the system and build an custom-designed business logic application for ACI Marine company which owns number of marines in Croatian Adriatic. The logic included custom multimedia graphical presentation of each marine (with VRML shows), on-line marine booking and custom on-line graphical interface for administration purposes. Figure 4. shows a presentation of ACI Marine “Opatija” with current availability status graphically presented. The graphical presented docks and icons are in realised as active web-elements so that functions can be preformed by an administrator simply by click of a mouse.

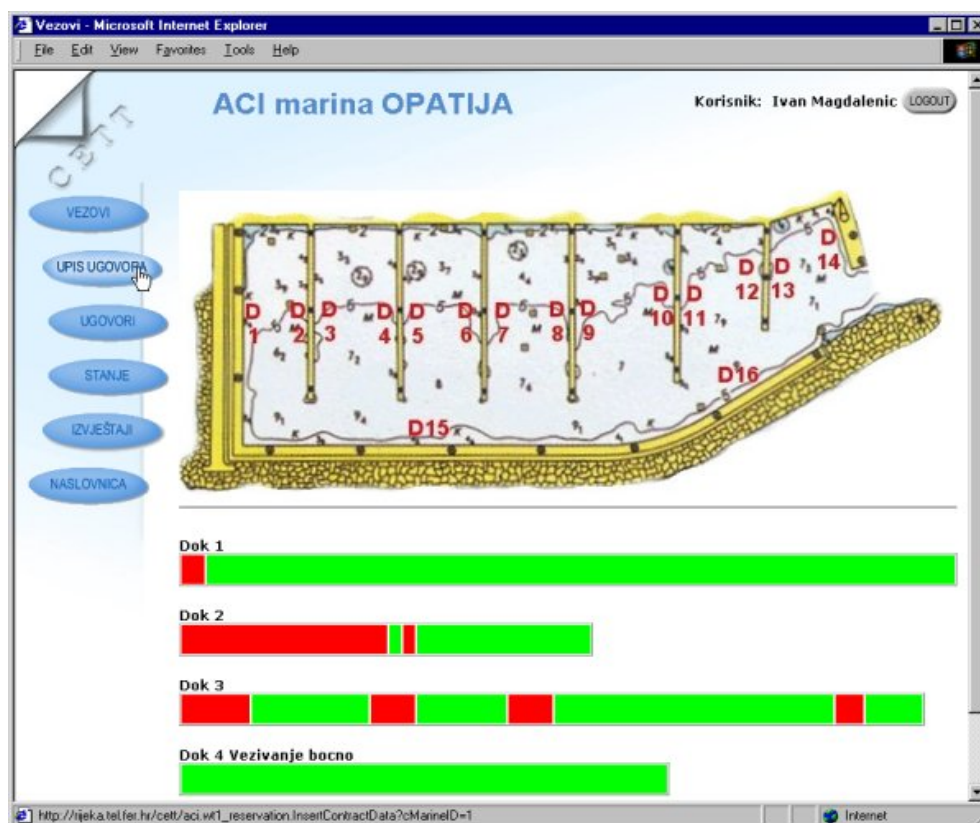


Figure 4 – Screen-shot of ACI Marine custom designed application

2.2 Multilanguage & Multicurrency

Europe is specific by diversity of languages and currencies. Small and medium size enterprises problem of a small market is even greater when they are surrounded with countries with different currencies and languages. Our goal is to develop a universal, simple and easy to manage solution. Unlike multilanguage solutions that are focused on storing translations in files or

making a mirror site for each language, we are storing translations in a database. The basic principle is shown in the Tables 1 and 2.

There are two tables, *Message* and *Translation*. *Message* is used for storing names of the messages used in code. They are replaced with the translation of the message (taken from *Translation*) into the customer's preferred language. Adding a new language to the application requires only adding new translations to the *Translation* table.

Multicurrency solution uses an analogous approach. There are two tables, *Currency* and *Exchange rate*. *Currency* is used for storing currency's multilanguage message name. *Exchange rate* store exchange rates against the default currency for every day. Product price is stored in the original currency and calculations to any other currency are made on demand.

Table 1: *Message table*

| Message_id | Message_name |
|------------|--------------|
| 1 | Greeting |
| 2 | Question_1 |

Table 2: *Translation table*

| Message_id | Language_id | Translation |
|------------|-------------|------------------|
| 1 | English | Hello! |
| 1 | Deutsch | Hallo! |
| 1 | Croatian | Bok! |
| 2 | English | How do you do? |
| 2 | Deutsch | Wie geht es dir? |
| 2 | Croatian | Kako si? |

3. DATABASE

As previously stated, all data is stored into the database, so no file system is needed. All business logic is also stored in the database in form of PLSQL procedures grouped in packages. Some of the packages grouping some of the business logic are:

- *Loose root hierarchy* (LRH) package - implements loose root hierarchy (each node can have one or more parent nodes, except top-level node),
- *Store* package - creates and changes data about stores, sections etc. Each store is implemented with LRH, each section presented as LRH node
- *Product List* package - implements list of products in a wishlist or a basket, includes product_ID, quantity and customer notes.
- *Basket* package – implements customers baskets.
- *Order* package – implements customers orders.
- *Product* package – manages products relating them to the section of the store.
- *Multilanguage* package – manages translation of messages and translation part of database
- *Account* package – manages system of user accounts.

Other packages include session control, credit card controls, transaction management, currency management, permissions and so on. A screen-shot of packages was shown in Figure 5.

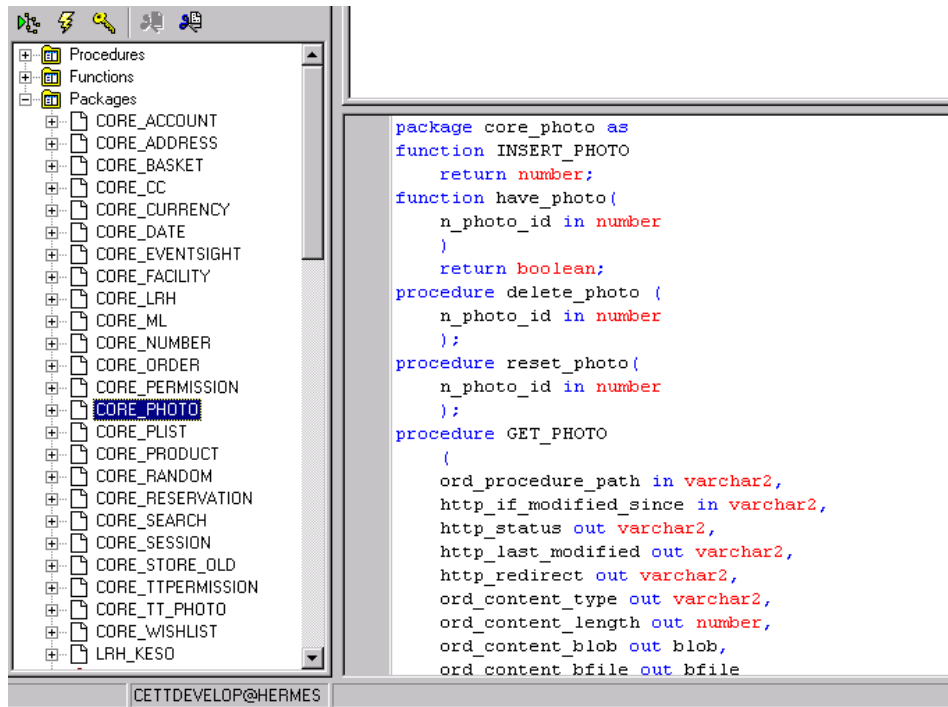


Figure 5 – Screen-shot of database Packages

Database schema includes 64 relational tables (some of them shown in Figure 6.). Database was modelled using Oracle Designer.

| Column | Data Type | Null? | Default |
|--------------------|----------------|-------|---------|
| ▶ PRODUCT_ID | NUMBER (12) | N | |
| CREATION_TIME | DATE | Y | |
| NAME_ML | NUMBER (12) | N | |
| PRICE | NUMBER (12,3) | N | |
| CURRENCY_ID | VARCHAR2 (3) | N | |
| MANUFACTURER_N... | VARCHAR2 (100) | Y | |
| STATUS | VARCHAR2 (12) | N | |
| QUANTITY | NUMBER (9,3) | Y | |
| MIN_ORDER_QUANT... | NUMBER (9,3) | Y | |
| QUANTITY_ROUND | NUMBER (9,3) | Y | |
| SECTION_ID | NUMBER (9) | N | |
| DESCRIPTION_ML | NUMBER (12) | Y | |
| PHOTO_ID | NUMBER (12) | Y | |

Figure 6 – Database Schema

CONCLUSION

The paper describes e-mall application residing on Oracle 8i database that provides small, medium and large enterprises to accept new modes of operation without the need to cope with technological and business issues that are related to these practices. The application aims at being key project that will led companies in adaptation to the business processes accepted in EU. The motivation behind the project of creating such an ambitious application is to establish a role model for transition of the country into networked economy.

Oracle 8i enabled storing all data and business logic into the database, so nothing has to be stored in file system. The system is more compact, open and easily expandable. Flexibility and modularity of the system using well-defined PLSQL procedures enables easy upgrades and new features incorporation into e-commerce application. All business logic and functionality of application is withheld within database and application server, so the client doesn't need to take care of any software.

CETT e-mall research program is setting up e-business support services and applications, becoming operative informational business centre that provides e-applications services via Internet. This way companies with limited human and other resources can benefit from B2B and B2C interactions.

Business layer is logically divided into three sections: tourism, goods and custom designed ACI Marine applications. Each of them uses principle of decomposition into four "ingredients": structure, content, look and functionality. **Structure** defines the purpose of every page and relations between the pages. **Content** ingredient is responsible for all the materials in the site perceived by the user as the "content". **Look** of the pages enwraps the content for a more pleasurable and effective human experience. **Functionality** captures all the specific processing involved with the site.

REFERENCES

- [1] Michael Stonebraker, Paul Brown: "Object-Relational DBMSs – Tracking the next great wave", Morgan Kaufmann Publiushers Inc., 1999.
- [2] Scott Urman: "Oracle 8i Advanced PL/SQL Programming", Osborne/McGraw-Hill Company, April 2000.
- [3] Kevin Loney, Marlene Theriault: "Oracle 8i DBA Handbook", Osborne/McGraw-Hill Company, October 1999.