Setting up e-Business Support Services in SME-prevailing Environments

E. Sehovic, I. Magdalenic and H. Simic

University of Zagreb, Faculty of Electrical Engineering and Computing (FER)
HR-10 000 Zagreb, Unska 3, Croatia
Phone: (+3851) 6129 763 Fax: (+3851) 6129 616 Email: enver.sehovic@fer.hr

Abstract

Large-scale adoption of e-business practices in SME-prevailing environments will be possible if appropriate support services are provided. Basic ideas of a development programs are presented, and preliminary results of an accompanying research project are exposed.

Keywords: Web, application hosting, XML

I. Introduction

Large-scale adoption of e-business practices is a complex task, particularly when Small and Medium-Size Enterprises (SMEs) are expected to play a substantial role in the development and restructuring of economies. Some SMEs will certainly be able to cope with technological and business issues related to these practices, but the vast majority will be not. If nothing is done, those companies will be left aside.

A research program, aimed at setting up e-business support services in SME-prevailing environments, had been launched. By re-allocating complexities in the system as a whole, which should be done through hosted services and applications, companies with limited human and other resources will be given the possibility to benefit from B2B, B2C and B2G interactions.

II. Objectives

The following guidelines have been adopted:

- There is a need for establishing e-business support services in Croatia, and for aligning the corresponding programs with those being launched in EU.

- e-business support services shall be offered to private and state-owned companies, government agencies, and individual customers.

- Services shall be conceived and deployed in such a way that both domestic and foreign users' requirements are met.

- These services and underlying facilities shall be linked to similar resources in EU, and shall become part of pan-European and global e-Business development programs.

More precisely, four tracks are taken into consideration. These are:

a) An electronic marketplace (exchange) for B2B transactions. Initially, this facility is expected to operate as a vertical exchange. All basic functions should be included (marketplace purchases, electronic catalogues, seller and buyer auctions, tenders, transaction delivery, online help). Later on, vertical exchange will evolve into a horizontal one. The potentials of the exchange will be gradually enhanced via value-added services.

b) A complementary electronic mall for B2C transactions. To be effective, the mall shall rely heavily on self-service functionality.

c) Application hosting based on "software-as-a-service" paradigm.

d) e-learning programs for speeding up the introduction of e-business services.

III. Development environment

Obviously, such a program cannot be restricted to academic community alone. Therefore, an effort is being made to organize a wider development environment consisting of:

a) A well equipped e-business lab at the

Faculty of Electrical Engineering and Computing (FER). FER is one of the two academic institutions from the whole EMEA region (Europe, Middle-East, Africa), which were selected for participation, within *Oracle Academic Initiative,* in a very ambitious *Oracle Applications Pilot.* FER also disposes with a small but functionally rich Public Key Infrastructure (PKI) for R&D.

b) Several large companies, which will carry on the program as a whole.

c) A network of co-operating companies, which will take care of auxiliary tasks (e.g. catalogue development, user training, fulfilment, etc.).

d) Financial institutions, which will provide e-business payment services.

The big picture being on the table, we shall present now some preliminary results For a better understanding of the text to follow, note should be taken that our initial research was focused on B2C issues. We looked primarily for ways how flexible electronic malls with abundant self-service functionality can be designed. The idea was to acquire experiences, which will fit in the more general B2C, B2B and B2G agenda.

IV. e-mall design principles

Standard three-tier architecture was adopted. Database tier, founded on *Oracle 8i* technology, is used for storing all business and platform related data (multimedia included).

The application is organized so as to be independent from the data storage. Business logic was conceived in compliance with Enterprise JavaBeans (EJB) architecture. EJB was chosen because of its robust, transparent and efficient storage and transaction management capabilities for enterprise-scaled applications.

XML is used in all communications (between application and external components, as well as between inner components of the application). Internal flexibility, required for prompt and localized updates and expansions, is secured.

XML documents, generated by business logic components, are input to XSLT, which, in turn, generates the corresponding user-oriented documents (usually XHTML).

a) Multilanguage & Multicurrency

If nothing is done, the diversity of languages and currencies will be a serious problem for small countries willing to play an active role on the e-business scene. A universal, simple, and easy to manage solution is required.

Instead of storing translations in files, or having a mirror site for each language, we stored the translations in a database. The basic principle is shown below:

message_id	message_name
1	Greeting
2	Question_1

Table 1: *Message* table

message_id	language_id	translation
1	English	Hello!
1	Croatian	Dobar dan!
2	English	How do you do?
2	Croatian	Kako si?

Table 2: *Translation* table

There are two tables: *Message* and *Translation*. *Message* is used for storing message names. Message contents, written in various languages, are stored in the *Translation* table. After the customer selects his or her preferred language, every message is replaced with the appropriate translation from the *Translation* table. When a new language is included, additional entries are inserted in the table.

Multicurrency is handled the same way. There are two tables: *Currency* and *Exchange rate.* Currencies' multilingual message names are stored in *Currency.* Exchange rates against default currency reside in the *Exchange rate* table. These data are updated daily. Product prices are stored in original currencies, and are recalculated on demand.

b) Business logic

From the technical point of view, business layer is modelled with EJBs which use Database Access Objects (DAO) for accessing database layer. DAOs encapsulate all of the relation's logic. If transition from one RDBMS to another is required, only DAOs are changed. The result of every EJB

action is an XML node, which contains formatted, ebXML-compliant, data. These nodes can be used in B2B interactions, or may represent parts of web page contents.

Business layer allows for flexible logical structuring of stores and items. The same item can be reached through various paths, as shown in Fig.1. This arrangement enhances search capabilities, and that might be quite convenient for the user.

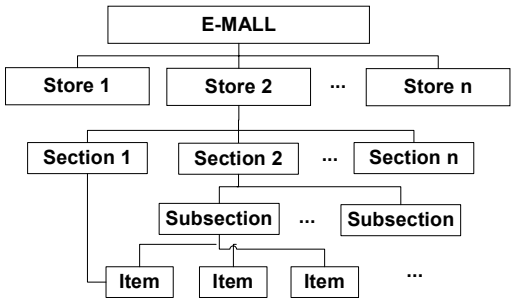

Figure 1: e-mall structure

V. Web site decomposition

Every dynamic Web site can be decomposed into four "ingredients": *structure, content, look* and *functionality*.

Ingredient elements can be collected into sets, and the site itself is defined by four such sets, one for each ingredient. If the site is to work properly, these sets must be mutually compatible. Interchangeable sets of the same ingredient allow for one "dimension" of the site to be changed, while the rest remains intact (e.g. visual redesign).

a) Site structure

In the Web layer, site structure is no more mapped to the file system directory structure. Instead, a tree residing in the data repository represents the site. Each node in the tree symbolises either a Web *page* (to display content), or a Web *action* (to trigger some application functionality and to return the link to the page that describes the outcome).

The structure tree in *extensive form* describes the site structure as it is seen from the outside (with URL paths). However, some nodes can represent multiple nodes, or even whole branches of the tree in the extensive form. These nodes can be linked to objects in the business layer. Figure 2 illustrates this relationship. The site structure is stored in *intensive form*, and is expanded at runtime.

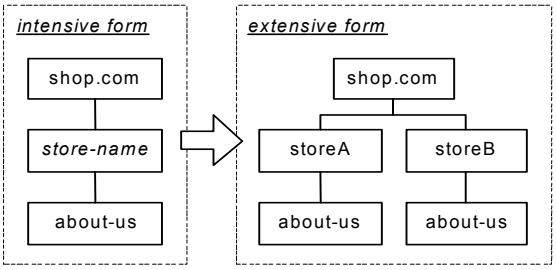

Figure 2: Intensive and extensive form of site structure

Pages may contain *page elements* of various types. These elements define the purpose of the page and can be inherited from a superior page. Thus, a more systematic and non-redundant description is obtained.

Content elements are one type of page elements. They collectively generate the content of the page in XML format, which can be delivered to the client as such (for inter-applications communication), or be transformed via XSLT into HTML (for human reading).

Special page elements are used for creating *page context*, mainly through inheritance.

b) Page look

The complete XML document created by the business layer is transformed with eXstensible Stylesheet Language Transformation (XSLT) into HTML. XSLT files are stored in the database. Before transformation, a pre-compilation of XSLT files is required to replace *xsl:include* tags with actual XSLT documents. With appropriate permission, online changes of XSLT files can be made through a Java applet.

VI. Data input mechanisms

Data input from the user is provided by two mechanisms: Web forms and Web navigation. Arbitrary textual and compound data can be filled in through Web forms and sent to the system in a single request. Navigation creates a page context, which can be also viewed as an input, although with a higher request-count-to-information ratio.

HTML forms are essentially presentation-oriented. They are mixed with

other HTML documents, which are used for defining the look of the page, and they provide no connection to business logic or data storage subsystems. HTML controls lack generic data structure.

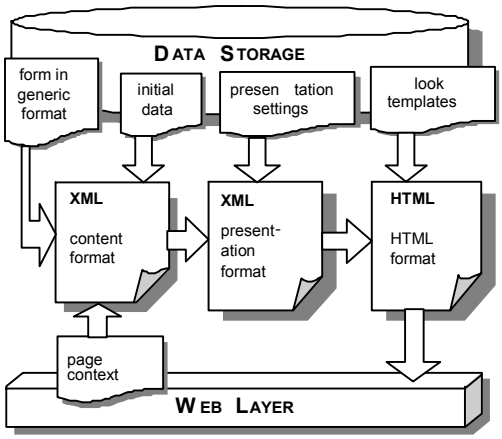

Figure 3: Web form generation procedure

Web layer specifies three new formats for describing Web forms, and it accepts HTML forms as the fourth format (as shown on Figure 3) Forms generation follows transformations from the first to the last format. In every stage information can be added or filtered out.

The first format is called *generic* because it uses generic types of items to describe the basic form structure. Multilanguage labels and constraints are also defined in this format. The construct is stored permanently.

The next format is *content* format, created as an XML document. Labels from the generic format are inserted in conjunction with the chosen language. Form items are assigned to initial data, if present.

The third format is used for *presentation* of high-level, generically typed, items. This XML document contains additional information about formats and control parameters. For example, multiple-choice lists can be implemented with select lists or checkboxes.

Finally, XSLT transforms the XML definitions of presentation format items (in charge of the page look) into a HTML code. The latter is sent as a response.

When complex data inputs are used in a transaction, multiple HTTP requests must be processed. In that case some resources will be locked and connections to the database will be open for a relatively long period of time. That can be improved if the transaction is started after all form fills have been gathered in a temporary data store. The processing code will view these data as if they were submitted from a single form.

VII. Key concept: application hosting

The crucial point is how SMEs can be assisted on their way to e-business. They will have to interact with individual consumers, other businesses and public administration.

B2C interactions will be facilitated if an e-mall service is provided, allowing SMEs to create and manage their stores in a very simple way (by using prefabricated templates and self-service functionality).

B2B interactions will be carried on most effectively if back office applications are delegated to and ASP, and selling and buying to an exchange. Further rationality will be obtained if these two services are merged.

B2G interactions will be possible when e-Government applications, relying on portal technology, become available.

VIII. Conclusion

Centralization of complexity and decentralization of information is one of the greatest achievements of Internet computing. For an end-user with a browser it is irrelevant who owns the server and the application: his/her company, or an ASP. Basically, this is a matter of price and service level agreement.

Implementation of hosted e-business support services is a chance and a challenge for small countries with numerous SMEs.

References

[1] L.Budin & alt: *Development Strategy of the Republic of Croatia: ICT,* Zagreb, April 2001, www.vlada.hr (in Croatian)

[2] W.D.Raisch: *The E-Marketplace,* McGraw-Hill, New York USA, 2001

[3] M. Hauswirth & alt: *An Architecture for Information Commerce Systems*, Proceedings of the ConTEL 2001 Conference, pp. 39-46, June 2001