eSarine - A Struts-based Webshop for Small and Medium-sized Enterprises

Nicolas Werro, Henrik Stormer, Daniel Frauchiger, Andreas Meier

Department of Informatics
University of Fribourg
Rue Faucigny 2
CH-1700 Fribourg
nicolas.werro@unifr.ch
henrik.stormer@unifr.ch
daniel.frauchiger@unifr.ch
andreas.meier@unifr.ch

Abstract: Electronic commerce becomes an important competitive advantage for small and medium-sized enterprises. Those enterprises therefore need a tool to easily exhibit and sell their products on the Internet. Electronic shops are the best solution to be accessible online but can be quite expensive. This paper presents eSarine, a Struts-based open source webshop, which is especially designed for small and medium-sized enterprises. eSarine features a number of advantages, among them a platform and database independency, a full internationalization support, the ability of handling several designs and a complete extensibility.

1 Motivation

In the last years electronic commerce has become increasingly popular. Today more than 87% of the companies are connected to the Internet, however only 16 to 22% of the small and medium-sized enterprises are exhibiting and selling their products online [Eu03]. Nowadays it is ever more important to be accessible online because we are moving from a traditional society to an information oriented one. In contrast to what has been thought in the '90s, electronic business will not replace traditional commerce but will be a vital complement of it.

When products are to be presented and sold on the Internet, electronic shops offer an interesting solution. Within a so called electronic storefront, customers can choose the products they are interested in, put them in a virtual cart and order those products online [Tu04]. Electronic shops also enable new ways of retrieving precisely information about offered products. Features like searching products with the use of key words, getting a detailed description of a product, showing all products of a special category and displaying product pictures can be found in almost all online shops. Those features, additionally to the fact that e-shops are available 24 hours a day, are undeniable advantages over the traditional remote ordering ways like the telephone or the fax.

Launching an online shop is expensive as different costs are involved. The most common costs are the software licenses for the electronic shop, the server to run the webshop, the connection costs to make the shop available on the Internet, the installation and adaptation costs of the software and finally the maintenance costs. In order to avoid the expenses of the software licenses and the adaptation costs, small and medium-sized enterprises can use open source e-shops which are available for free. Open source software is not only costless but also involves the publication of the complete source code of the software. This particularity makes the adaptation, installation and maintenance of the software possible at a very reasonable price if a minimum of IT skills is available within the company.

There already exist several open source webshop solutions like osCommerce[1], phpShop[2] and PHPepperShop[3] which provide the basic features needed to run a successful e-commerce web site. They all rely on the popular and powerful PHP scripting language which is widely supported by hosting companies [Me03]. Unfortunately PHP has also some drawbacks. The most significant one is that with PHP the presentation and the business logic are merged on every webpage. This particularity makes the code difficult to understand, to maintain and to extend for a third party enterprise.

This paper presents eSarine, an open source webshop that is not build on PHP but on Java which is actually the leading object-oriented programming language [Fl02]. Java allows programmers to develop either standalone applications, applets which can be embedded in a webpage or web applications with the help of Java Servlets and JavaServer Pages [HB04]. The choice of Java is motivated by the abundance of the Java platform, the support of the most recent technologies and the large Java community which provides open source Java developing and running environments.

This paper has the following structure: Section 2 introduces the Struts framework on which eSarine has been built. Section 3 explains the database connectivity approach chosen with eSarine. The different parts composing the eSarine webshop as well as their respective purpose are discussed in section 4. Finally section 5 gives the conclusion and an outlook.

2 Struts Framework

The eSarine online shop has been developed using the Struts framework provided by the Jakarta project of the Apache Software Foundation[4]. Struts is an open source Java-based framework which enables developers to create web solutions by providing them a set of functionalities. The core Struts technologies are JavaServer Pages (JSP) and Servlets used in connexion with JavaBeans, custom tags, XML, and web/application servers.

[1] Available at http://www.oscommerce.com (last visited 19/08/2004).
[2] Available at http://www.phpshop.org (last visited 19/08/2004).
[3] Available at http://www.phpeppershop.com (last visited 19/08/2004).
[4] Available at http://www.apache.org (last visited 19/08/2004).

2.1 Model-View-Controller

The entire Struts framework is based on the Model-View-Controller (MVC) pattern also referred as the Model 2 pattern [Ga95]. This pattern provides a clear separation between the controller, the model and the view of a system (see Figure 1). The controller, implemented as a Servlet, manages the application flow using the Command design pattern [Ga95]. The model, which can use JavaBeans (JB) or Enterprise JavaBeans (EJB), performs the business logic and encapsulates the application state. Finally the view consists of JSPs and custom tags to render the presentation of the result.

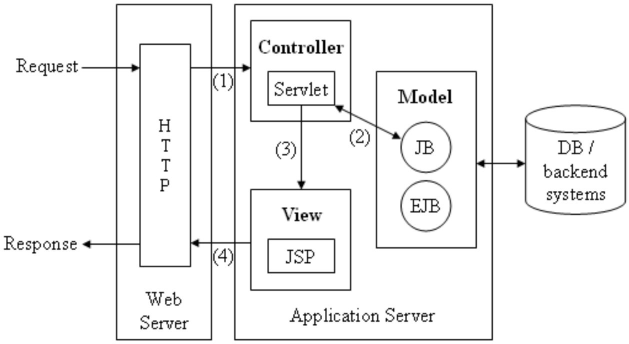


Figure 1: Implementation of the MVC pattern in the Struts framework

Whenever a request is sent to the server (1), the controller manages the workflow of the request processing. The controller first calls the proper model that contains the business logic to generate the result of the request (2). The control is then returned to the controller which dispatches the request result to the appropriate view for the presentation (3). The response is finally sent back to the user (4).

There are a number of advantages using the MVC pattern. The most important one is the complete independence of the model and the view as there is no business logic within the view part and no presentation logic in the model part. Additionally, different views representing the same model can be implemented and dynamically changed. In Struts the dispatching made by the controller to the different models and views is also completely independent and is defined in an XML configuration file. All those elements yield to a better component reuse and improve the ability to change the implementation of a certain layer with minimal effect on the other layers.

2.2 Internationalization

Another important feature supported by the Struts framework is the internationalization of applications. Internationalization enables an application to be available in different languages depending on the country the request came from. This mechanism is possible since browsers send with the requests a locale which is a lowercase two-letter code identifying the language, eventually followed by an uppercase two-letter code specifying the country or the dialect of that language, e.g. for English: en, en_US, en_GB.

For each language to be supported, a resource file containing a list of key/value pairs is necessary. The key is the identifier of the message and the value is the translation of this message in a given language. Then, instead of writing directly the text in the JSPs, a special tag with the key of the message is used and Struts will look for the appropriate translation regarding the locale. If no resource file is found for the given locale, the default language will be chosen. The default language is defined in a XML configuration file and is usually English.

Nowadays with the globalization, an internationalization mechanism is no more a strong but a required functionality. Struts easily allows web applications to be accessible for users coming from different countries and speaking different languages, increasing the number of potential customers.

2.3 Other Functionalities

The MVC pattern and the internationalization are the major functionalities of the Struts framework but there are a number of other functionalities that provide a significant help for developing web applications. As this article is not aimed to explain Struts in details, we will just mention some of the other important features. Further information on the Struts framework can be found in [Sp03] and [Hu03].

Struts implements session handling facilities to maintain the track of a user session between different requests. User session tracking is a recurrent problem with the HTTP protocol which can be resolved in two ways. The first one is to use the URL rewriting mechanism that simply adds to each hyperlink a so called ‘jsessionid’ key which uniquely identifies the customer. The second solution is to use a cookie containing the customer id to keep trace of the user. The Struts framework transparently uses the two methods; if cookies are accepted by the customer browser the session tracking will be achieved with a cookie otherwise the URL rewriting will be used. The first time a user accesses a web application, the server does not know whether the browser will accept a cookie, therefore the first response of the server will always use the URL rewriting method. This facility allows the implementation of a secured logon mechanism required in most web applications to maintain a public and a restricted area.

As described in subsection 2.1, views are made up of JSPs and custom tags. Struts offers its own set of custom tags which greatly simplify the creation of the JavaServer Pages. Some of those tags are counterparts of the standard HTML or JSP tags for mainly two reasons. First, the tags have to support the Struts features like internationalization and URL rewriting. A second reason is their ease of use and the clean coding approach they provide, keeping the JavaServer Pages readable. Struts also provides the notion of tiles which allows a JSP to be decomposed into several units of sense like header, menu, body and footer. With this principle, only the body has normally to be changed between the different pages making the maintenance of the presentation layer much easier.

Finally the exception handling mechanism is also a helpful feature of the Struts framework. It enables exceptions to be caught and correctly handles both the business logic layer and the presentation layer.

3 eSarine Database Connectivity

The most important component of a webshop is the data. The data contains information about the products, the different categories of products, the registered customers, the orders made and so on. The volume of the data can be large and there is a need to effectively and rapidly store, retrieve and update this data. For that purpose a DBMS (DataBase Management System) is the only efficient solution. For eSarine, being a Java-written e-shop, the connection to the DBMS can be seamless achieved with a JDBC driver often delivered by the DBMS vendor itself.

3.1 Database Schema

In order to store the information about products, categories, customers, orders and so forth, eSarine has a predefined normalized database schema reflecting all the webshop data and the system tables [SKS99]. This database schema has been developed to enable the addition of new system information such as product types, roles and languages discussed in section 4.

Very often related information like the products images and the resource files discussed in subsection 2.2 are not saved in the database but stored locally on the server. For several reasons, the eSarine has been designed to put all the information in the database by adding new tables to the database schema. The advantage is to enable a full web-based remote administration, disabling the need to use ftp (file transfer protocol) tools for managing images and resource files. Without this approach it would be quite troublesome to add, update and delete new products or new languages of the online shop. Another reason is to simplify the maintenance by eliminating the relative paths to the local directories containing the images and resource files.

3.2 eSarine DBMS Support

Many different DBMS are currently available on the market, some are open source other are commercial solutions. The objective of eSarine is to minimize the acquisition and maintenance costs for small and medium-sized enterprises. Therefore it is essential to support several DBMS. In the case of a company having a license for a DBMS a smart way will be to use the already installed system. On the other hand the best solution for a society without any database system license will be to install a free open source DBMS.

A significant problem among the different DBMS is that the syntax of their data definition language is not compatible. This assertion is particularly true for the data types definition and the call of legacy functions of the DBMS. This means that loading the correct JDBC driver is not enough to support a new DBMS. Instead, the implementation of a specific wrapper class, which encapsulates the specificities of the considered DBMS syntax, is required.

For the time being eSarine already supports the open source MySQL[1]and PostgreSQL[2] DBMS, the commercial Microsoft SQL Server[3], IBM DB2[4] and Oracle[5] DBMS. eSarine being itself an open source product, any programmer will be able to add new DBMS to this list. By supporting a wide range of DBMS, it is also possible to connect the eSarine webshop with an enterprise backend system by using the DBMS replication tools.

3.3 Connection Pool

Once the proper JDBC driver is loaded and the appropriate wrapper class has been instantiated, the eSarine webshop can create a connection with the DBMS. With this connection eSarine can execute query and update statements on the database. When the transaction is done, the connection has to be closed. The connection creation is time and resource consuming, therefore this approach is not efficient and would not allow many users to access the e-shop at the same time.

In order to be effective eSarine maintains and uses a pool of active connections. This pool is created at start-up time with a given minimum number of connections. If many requests arrive simultaneously, new connections can be created and added to the pool making the shop more scalable. The minimum and maximum numbers of connections are specified by the eSarine administrator and can be adapted over time.

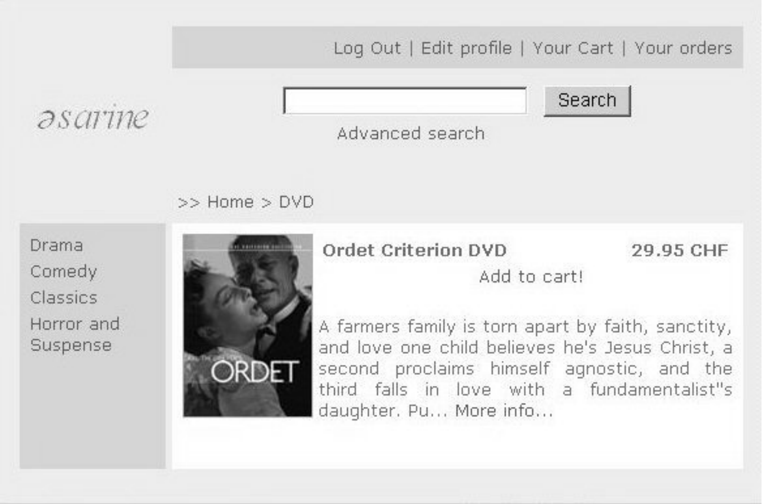
4 eSarine Features

Every webshop has at least two faces, a storefront part where the customers can browse and buy the products and a storeback part only accessible for the administrators where the e-shop can be administrated. The eSarine storeback allows the managers to remotely administrate the e-shop and is composed of the product, the language, the user, the payment and the shipping management.

4.1 Storefront

The storefront is the access point for the customers and therefore deserves a special attention as the first feeling is often the most important. The design of a shop is probably a key element of that feeling, therefore the default design of eSarine, shown in figure 2, has been kept pure and elegant. As nobody has exactly the same taste, eSarine easily allows the shop keeper to create a new design fitting better his will and the products he wants to sell. It is also possible to implement several designs and let the customers choose the one they prefer. Thanks to the MVC pattern of the Struts framework, the presentation layer is very flexible.

[1] Available at http://www.mysql.com (last visited 19/08/2004).
[2] Available at http://www.postgresql.org (last visited 19/08/2004).
[3] Available at http://www.microsoft.com/sql (last visited 19/08/2004).
[4] Available at http://www-306.ibm.com/software/data/db2 (last visited 19/08/2004).
[5] Available at http://www.oracle.com/database (last visited 19/08/2004).



Ordet Criterion DVD 29.95 CHF

Add to cart!

A farmers family is torn apart by faith, sanctity, and love one child believes he's Jesus Christ, a second proclaims himself agnostic, and the third falls in love with a fundamentalist"s daughter. Pu... More info...

Figure 2: The eSarine storefront within the DVD category

The ergonomics of a design is also crucial. For the default design we use the tiles functionality of the Struts framework by defining a header, a menu, a body, a location path and a footer. This leads to great component reuse as only the body is changing from page to page but, above all, improves the ergonomics of the website as the structure of the pages remains stable over the whole site. The customer will therefore quickly feel comfortable and easily navigate through the site.

The header tile is composed of the logo of the company, the actions the user can currently perform and the search field with a link for the advanced search. The menu contains the name of the categories or subcategories of the proposed products. The dynamic body tile can show a lot of different information like the products of the different categories, the detailed description of a product, the advanced search form, the register/login form, and so forth. On the top of the body the location path always indicates the category hierarchy the customer followed and allows him to directly go back to any parent category. The footer is actually empty but normally contains a string related to the trademark and legal information.

Another prominent aspect of online shops is the functionalities they offer. With eSarine a customer can basically browse through the categories of products, see the detailed description of a product, enlarge the picture of a product, search for a product with its name or, based on its attributes, in the advanced search, put products in the cart and edit the cart. With the help of the secured logon mechanism offered by Struts, we can allow an online customer to perform different actions depending if they are logged into the webshop or not. For example a customer who's not logged can either register if he's not yet a registered user or log himself in the system if he has already a username and a password. In contrast, when the customer is logged, which means that the shop uniquely identified him, the user is able to edit his profile, to order the items placed in the cart, to see the former orders he made and of course to log out.

The last point worth mentioning is the customizability of electronic shops. That's ever more relevant for online customers to be able to adapt some aspects of the webshops they regularly visit. The choice of the language is very important because it gives access to the information. eSarine, as we explained in subsection 2.2, checks the locale sent with the request and automatically displays the appropriate translation. This means that all the text within the pages like titles, menus, product descriptions and so on will be transparently customized. If the locale of the browser is not the correct one, which is often the case in the Internet access points, the customer can define in his profile a preferred language which will be applied after the login. Another useful customization provided by eSarine is the ability to define a foreign currency to allow the customers to additionaly see the corresponding prices in their own currency. As mentioned earlier in this subsection it is also possible to let the user define his preferred design or skin of the e-shop but, for maintenance reasons, not many shop keepers will offer this option.

The customization is only pertinent if the user can always and directly benefit from it. For this reason eSarine tries to set a cookie on the user browser so that the next time the customer accesses the online shop, he will be automatically identified and his preferences will directly be applied without having to log himself.

4.2 Product Management

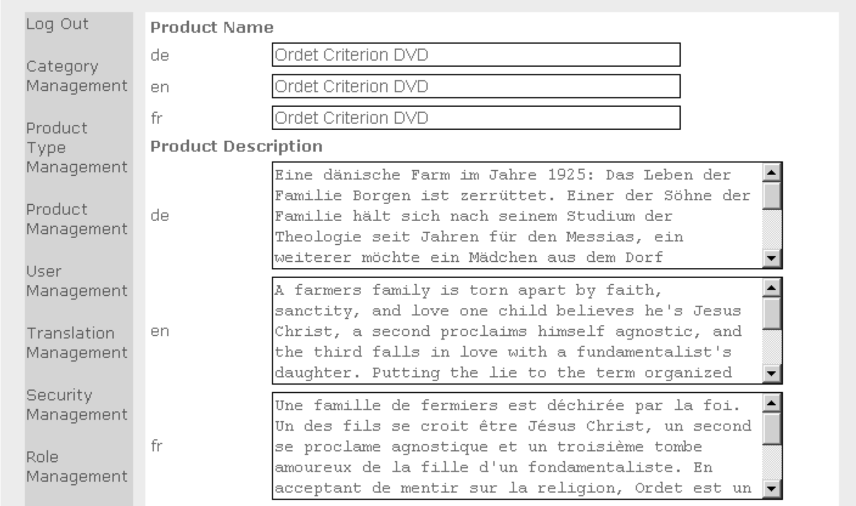
The webshop managers need a way to specify and to categorize the products they want to sell online. For that purpose, the concepts of category and product type have to be defined. eSarine forces every product to be part of a category. A category specifies a set of products and has a unique name, a discount and a parent category that can be the root category or any user-defined category. With the concept of product type eSarine gathers the similarities of the products. As each product has the minimal requirement of a name and a description, a product type specifies all the other common attributes a kind of products can have. For instance, the product type ‘book’ will declare the optional attributes title, author, year of publication, distributor and the ISBN number. A product type is itself defined with a unique name and a list of attributes. Every attribute of this list has a name, a field specifying if it is language dependent and two options indicating if it should be displayed in the detailed description and in the advanced search.

Once the categories and the product types have been defined the shop managers can insert the products. This is done by selecting the appropriate product type and then by giving the product a name, a description, a category and a value for all the additional attributes defined in the product type. Some more fields have still to be filled like the price, the discount and the available stock. To make the e-shop more attractive small and large images can be specified for each product.

As soon as a product has been inserted it will be available in the online shop. It can be accessed either by going to its category, or by a search. In both cases the product will first appear in a small description appearance (see Figure 2) which only displays the small image, the name, the price and the beginning of the description. By clicking on the ‘more info’ link, the detailed description will display the full description, the large picture and all the attributes with the show option activated.

4.3 Language Management

The language management gives the opportunity to add, edit and delete languages in order to internationalize the online shop. Note that eSarine using the Unicode standard can support almost all languages. There are two kinds of strings to be translated, the first one being the shop system strings and the second one being the attributes of all the dynamic elements added to the shop like the products and the categories.



Product Name

de Ordet Criterion DVD

en Ordet Criterion DVD

fr Ordet Criterion DVD

Product Description

de Eine dänische Farm im Jahre 1925: Das Leben der Familie Borgen ist zerrüttet. Einer der Söhne der Familie hält sich nach seinem Studium der Theologie seit Jahren für den Messias, ein weiterer möchte ein Mädchen aus dem Dorf

en A farmers family is torn apart by faith, sanctity, and love one child believes he's Jesus Christ, a second proclaims himself agnostic, and the third falls in love with a fundamentalist's daughter. Putting the lie to the term organized

fr Une famille de fermiers est déchirée par la foi. Un des fils se croit être Jésus Christ, un second se proclame agnostique et un troisième tombe amoureux de la fille d'un fondamentaliste. En acceptant de mentir sur la religion, Ordet est un

Figure 3: Internationalization of a product

As figure 3 shows, adding a new language to the e-shop is an easy but tedious task. You first have to go through the whole list of the system strings and give for each the correct translation in the newly added language. You then have to do the same with the language dependent attributes of every dynamic element already entered. For minimizing the impact of adding a language, it is highly recommended to choose the needed languages at the beginning in order to avoid the translation of the dynamic elements. To facilitate the task of the shop administrator, eSarine already supports three languages (English, German and French) so that the translations of the system strings are already done for those languages and only the dynamic elements will require a translation.

4.4 User Management

The user management enables to define ‘who is getting access to what’. As depicted in figure 4, the role-based access control used by eSarine specifies the entities user, role and resource to manage the access to the resources. To be flexible, users are not directly mapped to resources but to one or several roles which grant the access to different resources. The privileges of a user are then the addition of the resources allowed by his roles. eSarine provides two default roles which are ‘customer’ for the storefront access and ‘employee’ for full access. Managers have of course the possibility to change the permissions of those roles and to define new roles to implement a finer security policy.

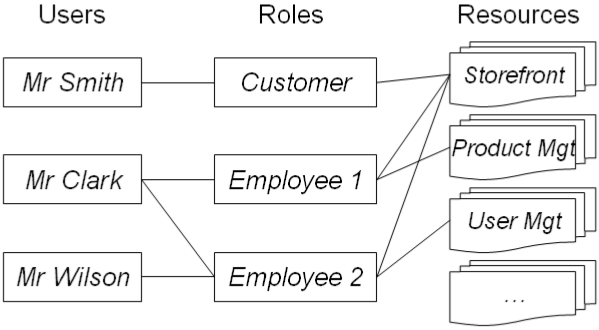


Figure 4: Example of a role-based access control

As an example take the role-based control represented in figure 4. Mr Smith being a customer can only access the webshop storefront. In contrast Mr Wilson has access to the product management part of the storeback and to the storefront. Mr Clark being at the same time employee 1 and 2 has in addition to the privileges of Mr Wilson the access to the user management part.

New users can be created in two ways: the first and the most common way is the registration of a new customer which implies the creation of a new user with the role ‘customer’. In addition, managers also have the possibility in the storeback part to create new users and assign them the appropriate roles.

The user management also includes the discount management. The final discount a customer will receive for a given product is calculated based on the discounts defined for the product, the category of the product, the user and also for the roles of the user. All the discounts are in percent and figure 5 shows how they can be combined to generate the final discount. This represents a very flexible discount management in which the shop administrators can define specific discounts on the products, categories and customized discounts for users and roles.

Discount	**Result**
user (R_u), role (R_{ri})	max(R_u, max(R_{ri}))
user (R_u), product (R_p)	R_u + R_p
user (R_u), category (R_c)	R_u + R_c
role (R_{ri}), product (R_p)	max(R_{ri}) + R_p
role (R_{ri}), category (R_c)	max(R_{ri}) + R_c
product (R_p), category (R_c)	max(R_p + R_c)
user (R_u), role (R_{ri}), product (R_p)	max(R_u, max(R_{ri})) + R_p
user (R_u), role (R_{ri}), category (R_c)	max(R_u, max(R_{ri})) + R_c
user (R_u), product (R_p), category (R_c)	R_u + max(R_p, R_c)
role (R_{ri}), product (R_p), category (R_c)	max(R_{ri}) + max(R_p, R_c)
user (R_u), role (R_{ri}), product (R_p), category (R_c)	max(R_u, max(R_{ri})) + max(R_p, R_c)

Table 5: The possible combinations of the discount types

4.5 Payment and Shipping Management

After the user has browsed the online shop and put some products in his cart, he can make an order. For that purpose eSarine supports different payment methods like the payment at the reception, the payment with credit cards and the payment with PayPal. Each method is implemented as a module making the addition of new payment methods an easy task. The credit card payment module is actually only registering the client's credit card information and the payment itself has to be done manually with a hardware equipment often available in small and medium-sized companies. An online credit card payment solution could be achieved through a financial gateway but as it needs special contract binding the society and the financial institution it was not possible to provide a generic solution. It can nevertheless be implemented as a new payment module following the instructions of the chosen financial gateway.

Once the payment has been executed, the order has to be shipped to the customer. Different shipping methods are possible depending on the nature of the product to be delivered. If the product is a piece of software it can be shipped online, for physical goods other shipping methods are needed. Therefore, eSarine also supports different shipping methods as modules. A shipping method has to calculate the shipping costs and for that purpose different approaches are possible. A simple solution would be to define a fixed price for all the deliveries. This approach could be extended by assigning free shipping for products with a certain price. Another way would be to calculate the costs based on the number of products ordered.

In eSarine, a weightpoint solution has been preferred where each product has a given number of weightpoints. If, for example, a shop sells books and chairs, books having a small size could get 1 or 2 weightpoints and chairs being heavier and much bigger could get 20 or 30 weightpoints. The shipping costs are then calculated by specifying for each shipping method the costs of the weightpoints regarding the region the delivery has to be done.

5 Conclusion and Outlook

eSarine is a Struts-based open source webshop which is itself totally cost free. The installation and the maintenance costs are reduced to the strict minimum as eSarine can be used with open source application environments including DBMS, developing environments and server applications. Low costs do not mean bad quality since eSarine relies on the Struts framework which provides a clean and robust software architecture. eSarine, being written in Java, is furthermore platform independent and easily extensible. By supporting the internationalization functionality, eSarine allows the online shop to be accessible for a very large number of customers around the world. These properties make eSarine especially interesting for the small and medium-sized enterprises which want to run into the breach of electronic business without engaging large financial assets.

As eSarine already supports the definition of several designs, a strong extension would be to support special designs for mobile devices like telephones and PDAs. Those devices becoming always more powerful they certainly represent the future of electronic commerce. Another important feature which is actually missing is a customer relationship management support to better interact with the customers and augment their loyalty [Ki03]. This feature can be implemented as a personalized mailing list, personalized offers and a personalized discount based on the real value of the customer. This value could be calculated with the help of a fuzzy classification [Me01]. These two proposed extensions will be implemented in the next months.

Bibliography

[Eu03] The European e-Business Market Watch: The European e-Business Report 2003 Edition – A portrait of e-Business in 15 sectors of the EU economy. Available at http://www.ebusiness-watch.org (last visited 20/05/2004).

[Ga95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[Fl02] Flanagan, D.: Java in a Nutshell – A Desktop Quick Reference. O'Reilly, Fourth Edition, 2002.

[HB04] Hall, M.; Brown, L.: Core Servlets and JavaServer Pages. Sun Microsystems, Second Edition, 2004.

[Hu03] Husted, T.; Dumoulin, C.; Franciscus, G.; Winterfeldt, D.: Struts in Action – Building Web Applications with the Leading Java Framework. Manning Publications, 2003.

[Ki03] Kincaid, J. W.: Customer Relationship Management – Getting it Right. Prentice Hall, 2003.

[Me01] Meier, A.; Savary, C.; Schindler, G.; Veryha, Y.: Database Schema with Fuzzy Classification and Classification Query Language: Proc. of the International Congress on Computational Intelligence – Methods and Applications. Bangor U.K., 2001.

[Me03] Meloni, J.: PHP Essentials – A Better Way to Learn Advanced PHP and PEAR. Premier Press, Second Edition, 2003.

[SKS99] Silberschatz, A.; Korth, H. F.; Sudarshan, S.: Database System Concepts. Mc Graw Hill, 1999.

[Sp03] Spielman, S.: The Struts Framework – Practical Guide for Java Programmers. Morgan Kaufmann Publishers, 2003.

[Tu04] Turban, E.; King, D.; Lee, J.; Viehland, D.: Electronic Commerce – A Managerial Perspective. Prentice Hall, 2004.