Volume 2, Issue 8, August 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Investigating the Web Application of AOP Using Aspect.Net Framework

Ms. Deepika Jangid* *Suresh Gyanvihar University, India*
Ms. Ruchi Dave *Suresh Gyanvihar University, India*

Abstract*— Aspect-oriented Programming (AOP) Methodology has been investigated in the design and implementation of a web application: Online music Store. The Aspect.NET Framework is implemented as an add-in to visual studio.NET. That means, the user can use Aspect.NET in combination to the integration development environment provided by Visual Studio.NET and its numerous comfortable features to develop software applications using AOP methodology. Some cross-cutting concerns have been identified and modularized into highly cohesive modular units-aspects, thus reducing the complexity of the design due to elimination of code scattering and tangling. The impact of using this methodology on various quality factors of the software has been examined. The study concludes that AOP methodology in Microsoft .NET platform and Aspect.NET framework can help in evolving efficient, cost-effective and quality web application.

Keywords— **Aspect-oriented Programming** (**AOP**), **Aspect**.**NET**, **Aspects**, **Cross-cutting Concerns**

I. INTRODUCTION

Aspect-oriented programming (AOP)[1] is programming based on identifying and creating of *aspects. It is a new emerging methodology that aims to improve the modularity and quality of software by achieving better separation of concerns.* The core construct of AOP is the aspect, which encapsulates behaviours affecting multiple classes into reusable modules.

Aspect-Oriented Programming (AOP) [1] complements OOP by providing another way of thinking about program structure. Whereas OO decomposes applications into a hierarchy of objects, AOP decomposes programs into aspects or concerns. This enables the modularization of concerns such as transaction management that would otherwise cut across multiple objects (such concerns are often termed crosscutting concerns).

Separation of Concerns is regarded as key principle of Software Engineering. It was introduced by Parnas and Dijkstra to manage the complexity of the software systems. In AOP the system is divided into two halves: the base (core) program and the aspect program. The base program contains the main functionality of the system and can be implemented using OOP methodology. On the other hand, the aspect program consists of the crosscutting functionality that has been modularized away from the base program. Crosscutting concern is a system wide concern whose implementation might affect several modules. Examples of cross-cutting concerns are logging, authorization, tracing, business policy implementations etc. Crosscutting concerns violate the modularization goal.

II. RELATED WORK

With today's modern Object-oriented languages like Java, C# or C++, ideally, a single concern can be modularized into a single unit that is a class, procedure or function, encapsulating all necessary structure and behaviour (called functional decomposition). But, unfortunately, there are some Crosscutting Concerns that cannot be captured with standard Object-Orientation's means of decomposition.

The most prominent and nearly ubiquitous example for this is simple logging which, necessarily, is present in most parts of an application and therefore crosscuts all logged classes or methods. Since all logging code is scattered across the application and tangled with the core functionality of each module, general modification of logging functionality can be a taunting task, as all locations where logging is applied have to be visited and changed separately. In more complex applications, additional Crosscutting Concerns, like security, transaction, performance, error handling or communication, have to be regarded, and one is left with code that is hard to comprehend, change and maintain.

So, the problem is of tangling of code that needs to be separated. And that can be done by separation of concern with following the AOP terminology. Modern AOP approach to software development is intended to solve a lot of issues related to increasing complexity of architecture, development and maintenance of software products. Aspect-oriented approach is helpful to simplify the business logic of an application, due to explicit separation of its cross-cutting concerns.

Theoretical foundations of AOP are well defined by a variety of researchers. However, even basic AOP concepts are still understood and interpreted different way by different researchers and developers. Except for widely known AOP tools for java- Aspect integrated into Eclipse, there are no AOP tools yet that could be easily integrated to the existing

software development environments. Aspect .NET Framework [2], the user oriented system, that provides a rich set of features to analyse and understand aspects and target applications subject to weaving. After review literature, it is found that Aspect .NET is still remained untouched for commercial applications.

III. USED METHODOLOGIES

This paper is focused mainly on Study of Aspect .NET framework [2] and Aspect oriented Programming in Aspect C# and then developing a web application (online music store) of Aspect Oriented Programming using Aspect .NET tool and Microsoft Visual C#. This investigation on web applications with AOP is done with the Aspect.Net framework. AOP is supported by java also but the Microsoft .NET platform is based on the principles of peer-to peer multi-language programming. For any of the .NET languages, a very comfortable toolkit for software development and maintenance is provided- Microsoft .NET framework and Visual Studio.NET. Basically aspect.net 2.0 is supportable with Microsoft's visual studio 2005.

IV. PROPOSED WORK

Having investigated all the issues affecting design of a website, a new proposal is developed to sort out the tangling of code. This is done by following AOP concept in Aspect.NET platform [2] which provides an integrated development environment and also provides a variety of code-behind language to choose from. So the work is initiated by designing a web application named online music CD store. In this website designing we have merged the concepts of aspects. Basically a website with functional features is designed. Then the aspects for cross-cutting concerns are designed separately. When the application runs and if there is a call of authentication or place_order aspect, then an automatic weaving of these aspects are done with the functional features. So here is the application online music CD store with its class diagram and aspect diagrams.

V. ONLINE MUSIC CD STORE

The objective of this project is to design a Music store web application with user interface which will enable them to browse, search, get song recommendations and buy the song items of their choice. This study focuses on modelling a representative 'Online Music CD Store' for the web users. A user/customer can browse and buy Music album or CD of his choice anytime on internet. The Fig. 1 below depicts the class diagram:

This Class diagram contains following aspects:

Logging: whenever any user going to login to music store. Than using logging aspect log file is created which records user's activity with date and time during login to logout session.

null textbox validation: when any of function from the functions placeorder, login, addcart_item is called and if the specified textbox contain null value than this aspect will display alert message.

Password_validation: when any of login function or registration function called than password_validation aspect checks if password field containing the length between 8-20. If it is not in this criteria than password_validation aspect shows alert message for it.

Place_order: whenever any customer places an order the same transaction page opens every time, so to reduce the same coding at each page, we can better make a dll for transaction that will be called at the pointcut and will be weaved every time at the join point.

By adding these aspects in the project it will be easier to handle those during transactions since the non-functional behaviour are compiled into aspects. So whenever they will be called by default they will be weaved with the functional modules. And we will not to see every time the thing like password is validated or not, logging is authorized or not.

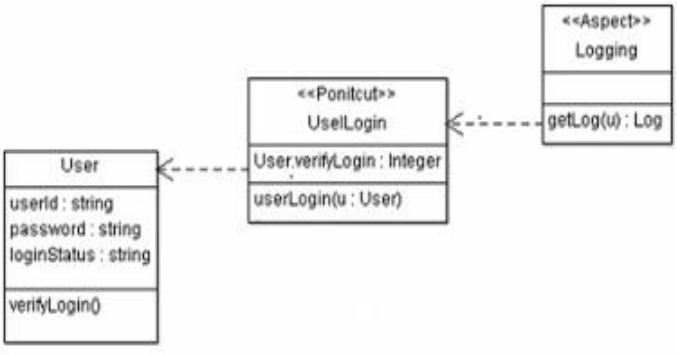

Fig. 2 Logging Aspect

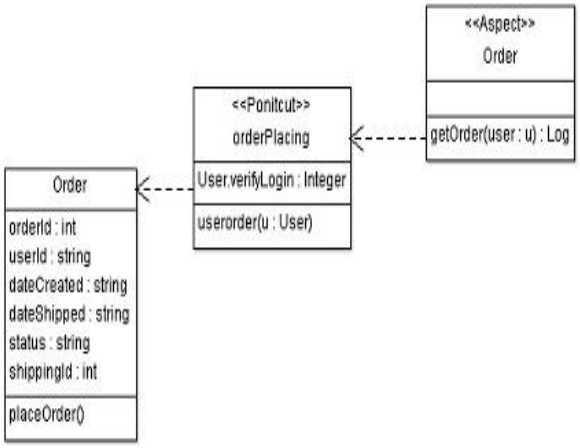

Fig. 3 PlaceOrder Aspect

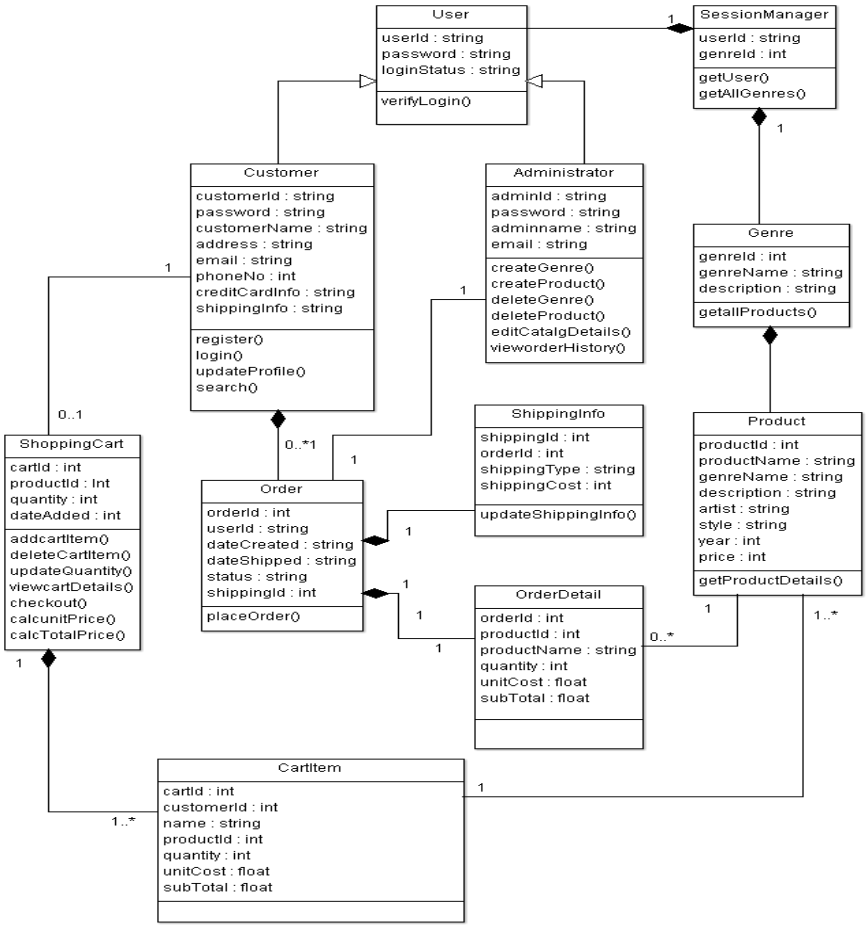

Fig. 1 Class Diagram of Online Music CD Store

Further, the repetitions of the code will be less. As we can see at every page of this project there are products to buy and for every transaction the same page will be opened but the product will be different. So at that time we can call upon the place_order aspect. That will make our task easier.

VI. CONCLUSION

It is apparent from this paper that implementation of Designing an application in aspect.net framework is much sorted out in comparison to object-oriented concepts. As it separates out the functional code and non-functional code, so the complexity reduces.

REFERENCES

[1] Aspect-oriented Programming http://www.en.wikipedia.org/wiki/Aspect-oriented programming

[2] Aspect Oriented Programming in .Net http://www.codeproject.com/KB/architecture/aop.aspx

[3] J.Hendrik Pfeiffer, John R.Gurd, "Visualization based tool Support for the Development of Aspect Oriented Programs", *School of Computer Science The University Manchester, Manchester.2005*

[4] Vladimir Safonov, Mikhail Gratchev, Dmitry Grigoryev, Alexander Maslennikov, "Aspect.NET-aspect-oriented toolkit for Microsoft .NET based on Phoenix and Whidbey" International Universitetsky prospect Petrodvorets, St. Petersburg, Russia

[5] Miguel Katrib Mora, Yamil Hernandez Saa, "Aspect Oriented programming in .Net based on attributes", Computer Science Department. University of Havana, PP-54-70-03/07, 2007.

[6] Amita Sharma, S.S. Sarangdevot, "Event Management System: Design and Implementation using AOP methodology In ECLIPSE-AJDT Environment", proc. In International Journal of Engineering Science and Technology (IJEST), PP-139-149, Jan 2011

[7] Amita Sharma, S.S. Sarangdevot, "Investigating the Application of AOP Methodology in Development of Product-Stock Maintenance and Billing Application Software using Eclipse-AJDT Environment", proc. In the International Conference *on Advanced Computing and Communication Technologie*, PP-446-450, 2011

[8] Vladimir O. Safonov, "Aspect .NET 2.1 User Guide", *St.Petersburg University*, 2007.

