Logger and Verifier: Security Patterns for E-commerce Application

Pallavi S. Kulkarni
Computer Department
Veermata Jijabai Technological Institute,
Matunga, Mumbai

B. B. Meshram
Computer Department
Veermata Jijabai Technological Institute,
Matunga, Mumbai

ABSTRACT
Design patterns, acting as recurring solutions to common problems, offer significant benefits such as avoiding unnecessary complexity, and promoting code reuse, maintainability and extensibility. Reusability and adapting to the changing software requirements are some of the challenges faced by software engineering community. In order to achieve these goals, application of Deign Patterns in software design is a proven practice. This paper describes the patterns which are useful in E-commerce Application. It also describes proposed patterns for E-commerce application with some modifications in the exiting Gang of Four patterns like Template method and FactoryMethod.

General Terms
Design Patterns,E-commerce

Keywords
Template Method,Logger,Verifier,FactoryMethod

1. INTRODUCTION
Design Patterns refer to reusable or repeatable solutions that aim to solve similar design problems during development process. Various design patterns are available to support development process [16][3][17]. Each pattern provides the main solution for particular problem. From developer's perspectives, design patterns produce a more maintainable design[2]. While from user's perspectives, the solutions provided by design patterns will enhance the usability of web applications [13][7][15][18].
Normally, developers use their own design for a given software requirement, which may be new each time they design for a similar requirement. Such method is time consuming and makes software maintenance more difficult. Adapting the software application to future requirements changes will be difficult if adequate care is not taken during design.[12]
Al local level patterns tell us how to solve particular problem for a given context. At the global level, patterns create a map how the components of the application interrelate with one another.[11]
Adopting Design Pattern while designing web applications can promote reusability and consistency of the web application. Without the adoption of the design patterns or wrong selection of them will make the development of web application more complex and hard to be maintained.[5][20]
Thus knowledge in design patterns is essential in selecting the most appropriate patterns in the web application.[10]
As modern e-commerce applications become very complex, the need for lower maintenance costs becomes more and more obvious. A part of this solution would be to develop the application in such a way, that it will be easy for someone to extend and further develop it[14]. The design patterns are one approach for solving this problem and they are applicable in object-oriented programming languages, such as Java. A design pattern is a well defined solution to a repeatedly occurring problem. From the programmer's point of view, a design pattern comprises a set of specific interactions that can be applied in common objects to solve a known problem.[3][4].
This paper is organized as follows. Section I is introduction which gives brief ides about design patterns. Section II focused on the design patterns which are useful in the E-commerce Applications. Section III discusses some of the patterns which are used for restructuring. Section IV discusses some restructuring applied to existing pattern to construct some new patterns for E-commerce Application. Paper concludes with section V.

2. PAGE SIZE
Web applications are normally build from scratch. Design patterns are providing reusability in the web application design. There are two types of patterns which are majorly used in the web application.

1. Interface Related Patterns – There are some patterns which are related to interface or GUI of the application. These patterns are like NEWs(Which displays the new updates),Opportunistic Linking (Giving option to user depending on the pervious history of the user)[22]

2. Hypermedia Application Related Patterns –this category of patterns deal with the behavioural, creational or structural patterns for the web application like MVC, State pattern, Session Pattern etc.[22]

All the patterns can be used with the application in different ways. Every pattern has some advantages and disadvantages.
While using the design patterns developer has to take proper care of the problem which is being solved. Sometimes design pattern usage increases complexity. For e.g If the problem can be solved using conditions, state pattern in such case may increase the complexity. So while applying the design patterns simplicity also should be taken care of.[10]
Some patterns are restructured or modified for the use in E-commerce application like Navigation Strategy, Decorator pattern.[1][18]

3. DISCUSSION
This section discusses various design pattern UML diagrams and structure of the pattern.

Patterns which are useful in E-commerce Application or Web application are
Creational Patterns: Abstract Factory, Factory method, Prototype, Singleton
Structural Patterns: Decorator
Behavioral Patterns: Command, Observer, Iterator, State, Session, Transaction, CallBack, Template Method, Strategy, Memento
System Pattern: Model-View-Controller

3.1 Factory Method

Factory Method is also creational Pattern which is useful to create one type of objects. It is used to define a standard method to create an object, apart from a constructor, but the decision of what kind of an object to create is left to subclasses

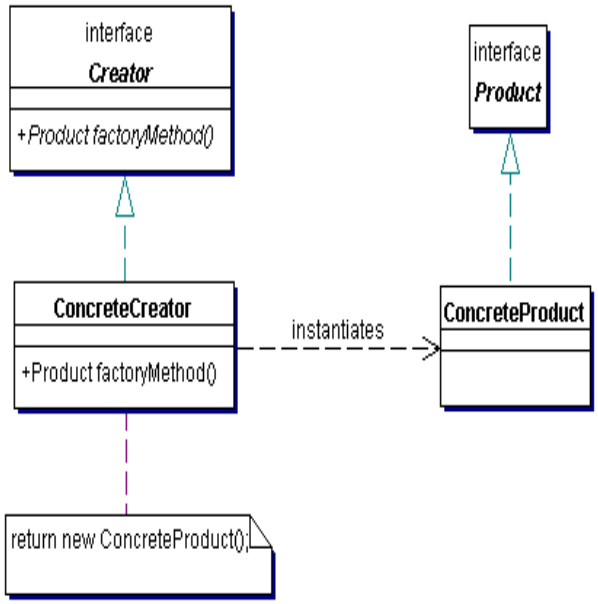

Fig 1: Class Diagram for Factory Method[3]

Product – The interface of objects created by the factory.
ConcreteProduct – The implementing class of Product. Objects of this class are created by the ConcreteCreator.
Creator – The interface that defines the factory method(s) (factoryMethod).
ConcreteCreator – The class that extends Creator and that provides an implementation for the factoryMethod. This can return any object that implements the Product interface.[3]

3.2 Template Method

This pattern is used to provide a method that allows subclasses to override parts of the method without rewriting it.
Use the Template Method pattern:
To provide a skeleton structure for a method, allowing subclasses to redefine specific parts of the method.
To centralize pieces of a method that are defined in all subtypes of a class, but which always have a small difference in each subclass.
To control which operations subclasses are required to override.[6][3]

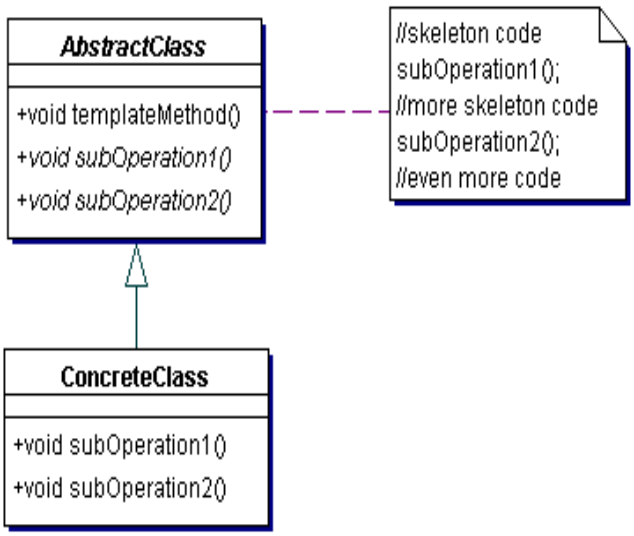

Fig 2 : Class Diagram for Template method[4]

AbstractClass – The AbstractClass is (perhaps not surprisingly) an abstract class that contains the template method and defines one or more abstract methods. The template method contains the skeletal code and calls one or more of the abstract methods. To prevent subclasses from overriding the template method it should be declared final.

ConcreteClass – The ConcreteClass extends the AbstractClass and implements the abstract methods of the AbstractClass. It relies on the AbstractClass to provide the structure of the operation contained in the template method.[3]

3.3. Singleton

This is object level pattern. Singleton is used when only one instance of the object is to be created. Some application requires a single object in the entire application. Singleton pattern allows the object to be created only once and then used by the entire application.Application: History List is the major application of Singleton. Another application is the Database Connection. For the entire application only one connection object is created and then that object is used.

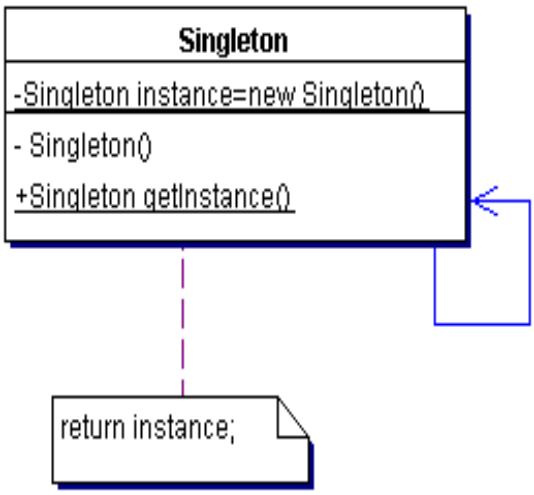

Fig 3: Class Diagram for Singelton[3]

Singleton – Provides a private constructor, maintains a private static reference to the single instance of this class, and provides a static accessor method to return a reference to the single instance.
The rest of the implementation of the Singleton class is normal. The static accessor method can make decisions about what kind of an instance to create, based on system properties or parameters passed into the accessor method.[3]

4. PROPOSED SYSTEM

E-commerce applications are normally three tier architures.Client, Business (Application) Layer and Database Layer.MVC is the pattern which is used extensively in web applications.[8][9][19]

The following diagram shows the growth of net commerce over the year

Fig 4: Growth of Net Commerce over the years.[21]

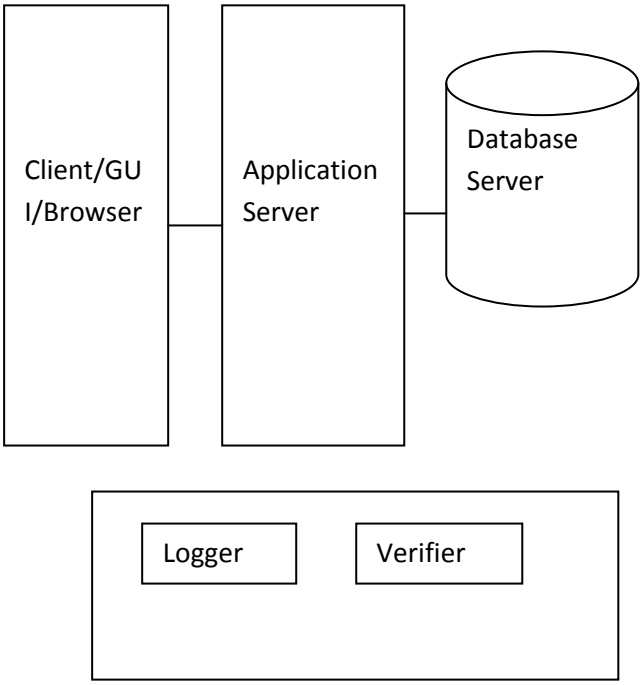

Fig 5: Proposed System Architecture

.
The application is divided into three different parts. View, Controller and Model are the three parts as shown in the above Figure 5. Controller part will use the existing pattern as well as the updated patterns.
View - View will have the GUI which has to be presented to the user. This part will work with observer, decorator[1], verifier patterns to create different view as per user role and for giving security by validating the input.
Model – This will have the database. This will work with verifier, transaction and observer pattern to perform the function of addition, deletion notification and committing the transaction. This is the database which is used to save the data. The data changes are handled by controller and reflected to view. This can be considered as database Module.
Controller – This will have the business logic. This is business Logic Module, which will have the patterns which are restructured for E-commerce Application such as logger, verifier.
As per the survey conducted by us for this work, we got the following results.

- Almost 80% people use E-commerce application for billing, purchasing online reservations or banking

things. We have many E-commerce sites are existing for the above mentioned things.

- The frequency of use of E-commerce site is almost 70% for above mentioned things.
- In the survey conducted almost 100% people was expressed the concern about the excpetions.As per the users and developers specified the requirement of the logging framework for exceptions.
- In the survey 98 % people said that there can be a common framework for verification and validation. User enters the information like shipping address or phone number. Administrator also enters the information about category, product. All this information has to be validated. A common solution can be used where just the information is passed in the form of the object and then validated. It will remove the redundancy code from the files and the standardized solution will be there for verification.

Following are the two patterns Logger and Verifier which are created by applying some of the restructuring to the existing pattern. According to Chikofsky and Cross[23], restructuring is defined as "the transformation from one representation form to another at the same relative abstraction level, while preserving the subject system's external behavior (functionality and semantics.)

4.1 Logger Pattern

Name of Pattern: – Logger Pattern

Type of Pattern: - Creational Pattern

Applicability: This pattern can be used for logging the exceptions in the E-commerce Application-commerce application have to be secure. Sometimes crash happens or some unwanted event happens and exception occurs. This exception is caught and some action is taken but this action is not recorded or the details or the exception is not recorded.

As the exceptions are different, one class is sufficient to handle all the exceptions. Here restructuring of FactoryMethod is done.

Using object oriented design principles we can have one interface which can be implemented by the logger abstract class. Singleton pattern along with Factory Method is used as logger pattern.

Refactoring used = "add class"

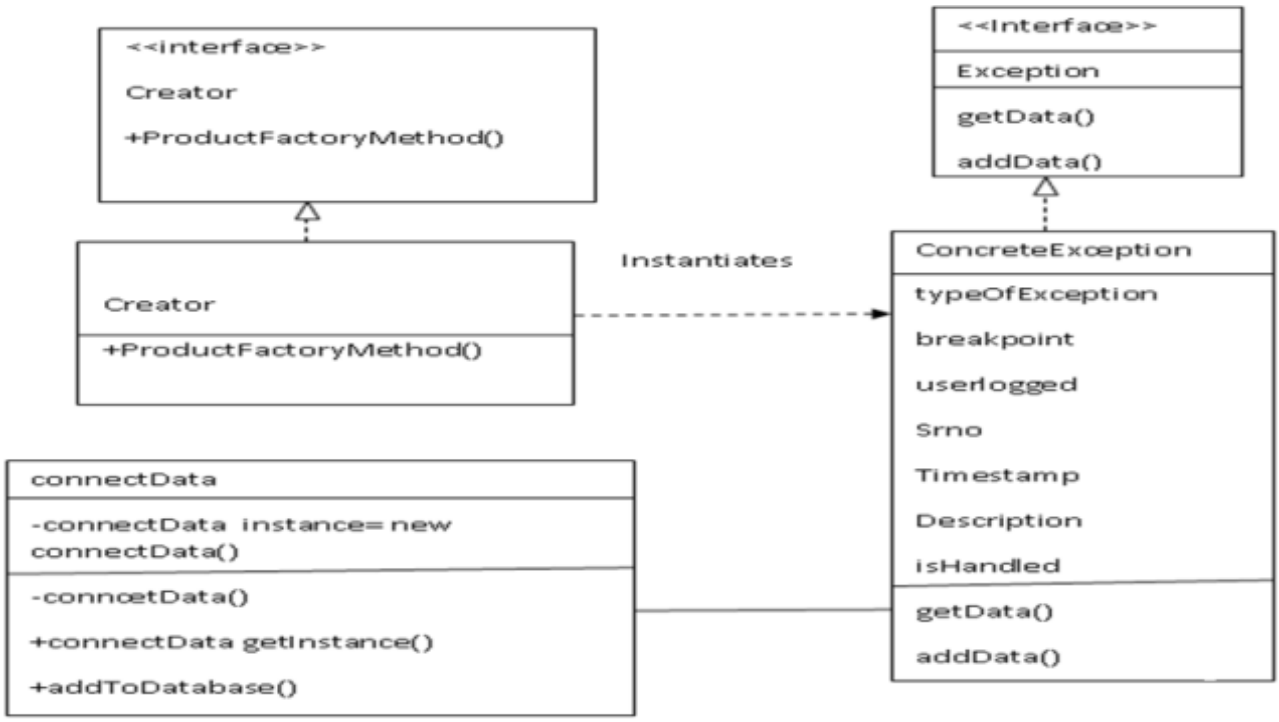

Fig 6: Class Diagram for Logger

- Creator interface provides the base for concrete Creator class to create the object for exception.
- Creator – This is the concrete implementation for the creator interface.
- ConcreteException will create the exception object to save that to database.
- Exception interface will have the methods for saving that to database.
- Singleton will be used here for creating a connection connectData.

The basic use of Logger pattern is to have crash reports. Logger pattern will prepare crash reports for security purpose. These crash reports can be analyzed further to identify the threats or the problems associated with the application.
As per the survey 100% people wants the exceptions to be in readable form like in which form the exception occurred with which object it is occurred and some information which will help to protect the site from attacks which are generating exceptions and hampering the performance of the E-commerce sites likes SQL injections or entering invalid information to get the details of the SQL table structures.
These crash reports can act as a analysis tool for the security for the site.

4.2 Verifier Pattern

Name of Pattern: – Verifier Pattern

Type of Pattern: - Behavioral Pattern

Applicability: Whenever data is entered into the system, verifier gets that data and validates it according to its requirement. Here strategy pattern will have the object as the parameter when it gets the requirement of validation and the data entered. This will verify the data and gives the report to the user.

This pattern internally uses the Template Method pattern with the refactoring "Design by contract".
This pattern can be used along with any pattern where we require validating the user input. As the validations differ from each other in many ways like alphabet validation, alphanumeric validation, special character validation, so instead of writing different classes, one class can handle all the requests.

The validation request will be given to this class in the format of object. The refactoring applied here is "convert parameter to Object".

The processing will be done and the result will be displayed to the user and data will be validated and processed.

- validateBehaviour – This interface will generate the object for verifier.
- Concreteclass – This class will have the concrete implementation of the method of template interface
- Template Method Interface– This will have the interface which is providing some of the common methods for verification.

Refactoring Used: Design by Contract and Convert Parameter to Object.
Refactoring Design by contract means the data is validated before entered into the database at the border of the system.Refactoring Covert parameter to Object means when the method has many parameters to be passed to it, all the parameters are embedding into one object and then the object is passed to the method.
Verifier will help the developer to verify the data of different type like character ,alphanumeric, date numbers etc.As we are passing everything as a parameter to the program as a single object so no need to write different function for every verification. One single function will have every type of validation so it will have good cohesion in the system.

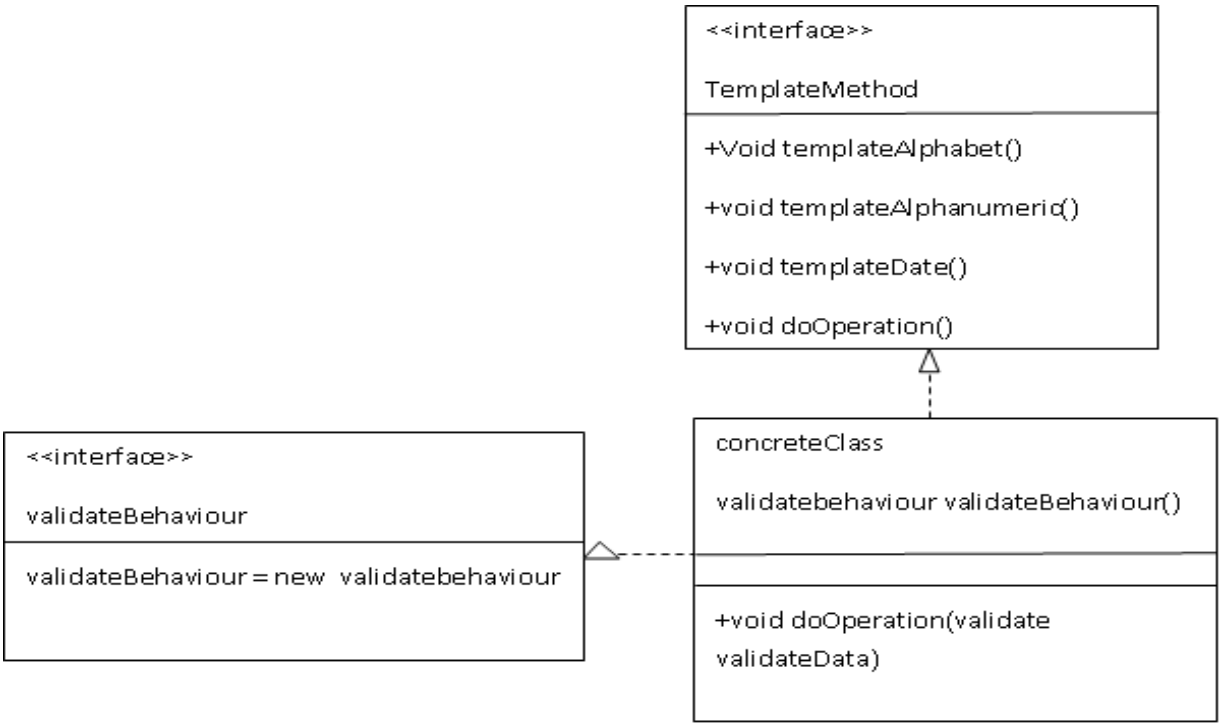

Fig 7: Class Diagram For Verifier

5. CONCLUSION

Many of the existing design patterns are used in the E-commerce application in association with other patterns. Mainly MVC pattern is used in E-commerce application. Pattern has some draw backs. Some variations can be added to patterns to make it efficient. Design patterns are proven solution to re-occurring patterns.

The discussed patterns are used for security problems associated with the E-commerce applications. They are derived from the existing proven patterns.

Logger pattern will help the administrator to view the crash reports and analyze the security aspect. Verifier will help to verify the information with some common structure with parameter as object. The patterns are solving some occurring problems related to security.

6. REFERENCES

[1] Vijay K Kerji ,"Decorator Pattern with XML in Web Application", ©2011 IEEE

[2] Jaeyong Park,David C. Rine,Elizabeth White, "Assessing Conformance of Pattern-based Design in UML", 2008 ACM

[3] Stephen Stelting,Olav Maassen,"Applied Java™ Patterns", ,Publisher: Prentice Hall PTR First Edition December 01, 2001

[4] Eric freeman and Elisabeth Freeman ,"Head first design patterns",Orielly publication,

[5] Phek Lan Thung, Chu Jian Ng, Swee Jing Thung, Shahida Sulaiman,"Improving a Web Application Using Design Patterns: A Case Study", ©2010 IEEE

[6] Michiaki Tatsubori Toyotaro Suzumura, "HTML Templates that Fly A Template Engine Approach to Automated Offloading from Server to Client", WWW 2009 MADRID! Track: Web Engineering / Session: Client Side Web Engineering

[7] Gustavo Rossi,Fernando Lyardet, Daniel Schwabe ,"Patterns for E-commerce applications",

[8] Patrick Sauter Æ Gabriel Vo¨ gler Æ Gu¨ nther Specht Thomas Flor,"A Model–View–Controller extension for pervasive multi-client user interfaces", _ Springer-Verlag London Limited 2004

[9] Maria Mouratidou, Vassilios Lourdas, Alexander Chatzigeorgiou and Christos K. Georgiadis,"An Assessment of Design Patterns' Influenceon a Java-based E-Commerce Application",

[9] Alan shalloway,"Can Patterns be Harmful", ,Cutter IT Journal,@net Objectives

[10] Steve Macdonald,Kai Tan,Jonathan Schaeffer amd Duane Szafron, "Deferring Design Pattern Decisions and Automating Structural Pattern Changes Using a Design-Pattern-Based Programming System",ACM Transactions on Programming Languages and Systems, Vol. 31, No. 3, Article 9, Pub. date: April 2009.

[11] Allan Shalloway,James R.Trotta ,"The Principles and Strategies of Design Patterns",Net Objectives,Volume 1,Issue 4,April 2004

[12] Paloma Díaz,Ignacio Aedo,Mary Beth Rosson ,"Visual representation of web design patterns for end-users,",Copyright 2008 ACM 1-978-60558-141-5

[13] Alejandra Garrido, Gustavo Rossi and Daniel Schwabe ,"Pattern Systems for Hypermedia",

[14]Peng Li,Manghui Tu · I-Ling Yen,Zhonghang Xia,"Preference update for e-commerce applications: Model language, and processing", Springer, Electron Commerce Res (2007) 7:17–44 DOI 10.1007

[15] Giuseppe Antonio Di Lucca, Anna Rita Fasolino, Porfirio Tramontana, ,"Recovering Interaction Design Patterns in Web Applications", Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05) 1534-5351/05 © 2005 IEEE

[16] Jeffrey Heer and Maneesh Agrawala, "Software Design Patterns for Information Visualization", IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 12, NO. 5, SEPTEMBER/OCTOBER 2006

[17] Bruce Eckel ,"Thinking in Patterns"

[18] Gustavo Rossi, Daniel Schwabe, Fernando Lyardet , "Improving Web Information Systems with Navigational Patterns"

[19] Avraham Leff, James T. Rayfield ,"Web-Application Development Using the ModelNiewlController Design Pattern",0-7695-1345-2UOl/$lO.O0O 2001 IEEE

[20] T.H. Ng,S.C. Cheung,W.K. Chan, Y.T. Yu, "Work Experience versus Refactoring to Design Patterns:A Controlled Experiment".

[21]http://www.imediaconnection.in/article/91/Digital/Internet/ecommerce-in-india-statistics-trends-and-insights.html

[22] Pallavi S. Kulkarni,pradnya Rane,Suchita Patil,

Dr.B.B.Meshram" Use of Design Patterns in E-commerce Application:Survey", International Journal of Advanced Research in Computer Science, ISSN: 0976-5697, Volume 3, No. 1, Jan-Feb 2012

[23] Tom Mens, Tom Tourwe. "A Survey of Software Refactoring", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 2, FEBRUARY 2004