

Analyzing the Impact of Software Design Patterns in Data Mining Application

Nair Puja Prabhakar, Devika Rani, A.G. Hari Narayanan
and M.V. Judy

Abstract Data Mining is all about analyzing the data gathered from different sources, based on different context and summarizing it into meaningful information. The belief of utilizing design pattern to improve the data mining application is relatively high. Design Pattern is a broad and general repeatable solution to a problem that occurs frequently in a software design. There are only a few number of researches illustrating their benefit. In this paper, we try to expose the relation between data mining architecture and design patterns. We have here taken a layered architecture for data mining environment and analyzed the impact of design pattern in every components of given architecture. This paper presents a survey on various design patterns used in data mining which is used to fulfill functional and non-functional requirements (quality attributes).

Keywords Data mining · Design patterns · Layered architecture · Quality attributes

1 Introduction

We are living in the information age and thus a large amount of data is gathered on daily basis. The important need is to analyze such data. Data mining is used to turn such large collection of data into meaningful knowledge. Data mining is completely

N.P. Prabhakar (✉) · D. Rani · A.G. Hari Narayanan · M.V. Judy
Department of CS and IT, Amrita School of Arts and Sciences,
Amrita Viswa Vidyapeetham, Amrita University, Kochi, India
e-mail: pujaprabhakaran@gmail.com

D. Rani
e-mail: devikahari07@gmail.com

A.G. Hari Narayanan
e-mail: hariag2002@gmail.com

M.V. Judy
e-mail: judy.nair@gmail.com

an interdisciplinary area that can be explained in different ways. It is a process of examining data from many different perspective or dimensions and then summarizing the identified result. The various data mining application include marketing and fraud detection. This marketing contains customer detailing and retaining, identifying probable customer, market partitioning, and fraud detection is all about identifying fraud in credit card, intrusion detection. It can be viewed as an essential step in the process of knowledge discovery, where other steps include the following:

1. Data Cleaning
2. Data Integration
3. Data Selection
4. Data Transformation
5. Data Mining
6. Pattern evaluation
7. Knowledge presentation

In software, a design pattern is a general reiterative solution to a commonly existing problem in software design. They are not a finished design that can be applied directly to code. In many different situations, they are the fingerprint or impression for solving the same.

Commonly, a pattern has four essential elements, which are listed below:

1. The **pattern name** is a controller which can be used to describe a design problem, where it occurs and its solution in a word.
2. The **problem** describes the context where the given pattern can be applied.
3. The **solution** is used to explain the element that forms a design domain, their work flow, collaboration among each design.
4. The **consequences** are the outcome or result that we get after applying the pattern.

An efficient design but if it takes an hour to generate a solution then it is of no use to the person who needs result at that moment. Carefully developed and well-defined designs are the foundation of better engineering. Design pattern is a good option for maintainability, as it provides a new means for reusing existing designs and architecture. Understanding the present techniques through design patterns makes them more accessible to developers of new architecture. They help you to choose design option in such a way that makes a system reusable and avoid options that do not support reusability. With a genuine design pattern we can thus improve the maintainability. So we can say this design pattern describes the problem which has a chance to occur again and again in given environment, and then define the core of the solution in such a way that they can be used million times repeatedly without ever doing the same thing again. Here we have taken a layered architecture of data mining environment and analyzed the impact of design pattern in the different components of this architecture. Through this way Data mining algorithms' adaptability and maintainability can be improved.

2 Related Work

“Acceptor–Connector an Object Creational Pattern for Connecting and Initializing Communication Services”. In this paper, they discuss about the acceptor–Connector pattern, when a service is initiated this design pattern decouples connection generation and service initialization from the already performed processing. To perform this decoupling here three components are used: (1) acceptors, (2) connectors, (3) service handlers. A connector powerfully produces a connection with acceptor, and a service handler is initialized in order to produce a data exchange. Where an acceptor continuously waits to get request for connection, when such request is received the connection is established and data are exchanged.

Benefits provided by such patterns are the following:

1. Reusability is achieved
2. Maintains portability
3. Extensibility is achieved

So here they have mentioned the benefits of using such pattern in distributed environment [1].

“A Pattern-Based Data Mining Approach” Patterns are very useful in software design as the well known, GOF pattern for OO design pattern. Pattern can be applied to much wider context, with the creation of such pattern in various fields it seems data mining algorithms’ adaptability and maintenance can be improved. A pattern can be identified as the best one through the way they generalize and solve the actual problems. The quality and standard of a design pattern can change over time when new one gains acceptance or better solutions are available. This paper gives a brief introduction about different design pattern available in data mining. They have already identified a collection of other design patterns in the area of data mining, which are listed below:

1. Combine Voting
2. Training/retraining
3. Solution analysis

Pattern can be applied to any area of human interest. Through this pattern, we will be able to give users the possibility to model data mining, and also to better understand the data mining step [2].

“A Pattern-Based Approach for GUI modeling and Testing” In this paper, they have discussed about a pattern-based approach for GUI modeling and testing. User Interface (UI) patterns are used widely in the software design. This UI patterns exhibit commonly occurring solution to solve common UI problems. GUI is the fundamental part in software, which makes the software user friendly by offering flexibility in how user tasks are performed. They have made two observations: (1) Developers use distinct tools to benefit the creation of GUI. (2) But, it is a fact that these tools and toolkits are only able to support the beginning phases of GUI’s

development steps. To support the above two observation, the specific tool is used for user interface patterns [3].

“A Survey on Software Design Pattern Tools for Pattern Selection and Implementation” This paper presents a survey on various methods to select the suitable design pattern for problem in hand. It further discusses the different tools that are available to apply this design pattern into code. They ended the paper saying that in the upcoming years software design knowledge can be observed in the form of strategic and tactical patterns that solve the problem with various client/server programming, distributed analysis, software maintainability, real-time applications, embedded system, game creation, and user interface design [4].

“A Software Architecture for Data Mining Environment” The technological effectiveness, predominant the use of paradigm, are a major factor that influences the acceptance and the impartiality of such background. They have here addressed issues such as integration, where new tools can be added into environment, interoperability over numerous dimensions and the interaction with other framework available, other issue is agreement between legacy tools while choke in the environment and its accommodation with the interoperability components which is enforced [5].

“A Comprehensive Approach Toward Data Preprocessing Techniques and Association Rules”. Data mining is a technique that details with big data in that preprocessing plays an important role in this mining process and huge contribution in success of project. This is also useful for association rule algorithm. In this paper, they have processed a methodology and implementation of data preprocessing and then mined the rule with prevailing association rule algorithm [6].

3 Data Mining Architecture with Design Pattern

Software architecture is a general topic which gives less concern about implementation part than to the design. It usually includes the concept necessary to describe the structure and concepts, lighting toward its evaluation. Our description is based on the fact that how this architecture explains the structure or the structure of this data mining application. The architecture that we have taken here is layered one and can be described as three layered architecture which is mentioned below:

1. User interface, which the user has interaction with, it is first thing to which user puts attention and acts as a bridge between user and business components.
2. Business components is the layer that takes care of processing tools and preprocessing part. This is basically a two group layer.

The storage component and connectors are used to store all the data that are gathered and maintained which need to be manipulated later in the journey of data analyzing (Fig. 1).

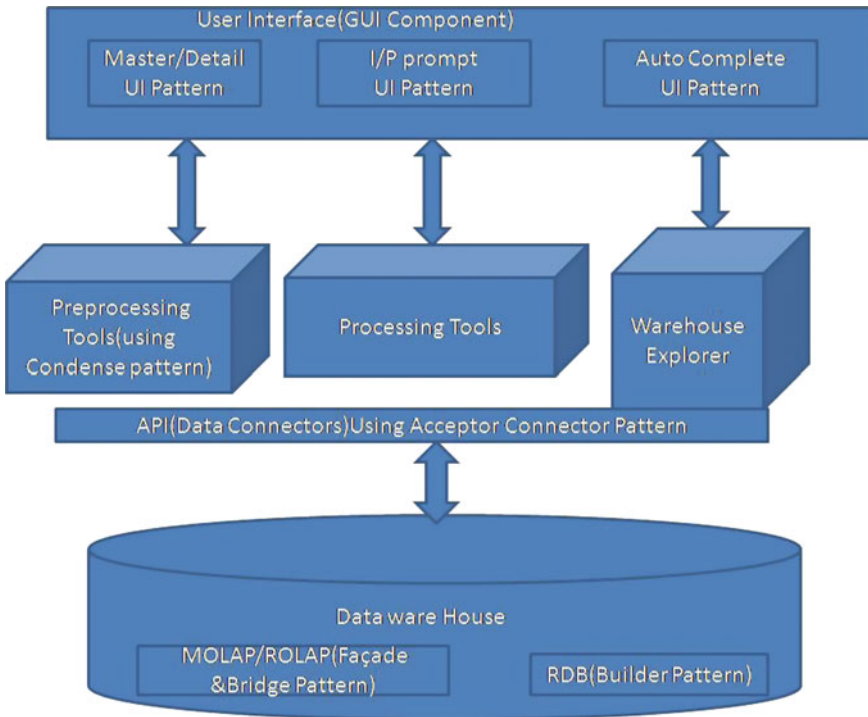


Fig. 1 Design patterns used in the layered architecture for data mining environment

User interface is a detailed graphical implementation of certain functionality, but the fact is layout may be different based on functionality, but our work is based on common behavior that is repeated over time. To get an entry, we researched on different user interface design patterns in data mining.

The Master/Detail UI Pattern is the one that divides the screen, into two areas master and detail area, based upon the selection made in the master area, the content in the detail area is displayed.

The Input Prompt UI Pattern allows the user to enter data into the computing system.

The Auto complete UI Pattern allows a word-based searching which means when a user inputs a word, a list of all possible outcomes appears and finally desired one can be selected [3].

The next component is business component which contains two parts: first is preprocessing tool and other one is processing tool. The preprocessing tool is used to ensure data quality which has various factors like accuracy, completeness, consistency. The preprocessing tools like filters use patterns like condense pattern, which is one of the most known data mining pattern [2]. This condense pattern is basically used as preprocessing pattern.

The third component present in this architecture is storage component and connectors example data warehouse. Data warehouse provides various tools to administrator for organizing the data methodically and use this data to make better decision. There are four major keywords related to data warehouse, which are subject-oriented, time variant, integrated, and nonvolatile collection of data. This data warehouse can be viewed as multitier architecture; data warehouse is usually viewed as a two-tier architecture where the bottom tier is warehouse database server which is relational database system used as a back-end tool. Which can considered as an example of Builder pattern which comes under creational pattern, builder pattern is the one that separate the development of a complex object from its embodiment so that the same development process can create different embodiment.

The next tier is OLAP server which is consistently implemented using one of the following: (1) Relational OLAP (ROLAP) model or (2) a multidimensional OLAP (MOLAP). Here we can use either of the following patterns bridge or façade, which comes under structural pattern, this pattern is bothered with how classes and objects are created to form bigger structure, this pattern is useful for making independently created class library work with each other. The various OLAP operations like Drill up/Drill down can be considered as an example of iterator design pattern; they provide a means to access the components of an aggregate object in a successive order without disclosing the underlying representation.

Data loading that takes place in data ware house or data mart can be considered as an example of mediator pattern, which is a behavioral pattern, this pattern is used to define how the collection of objects interact.

4 Impact of Design Patterns in Achieving the Quality Attributes of Data Mining Application

There are various quality attributes that has to be achieved in data mining projects. Some of the prominent quality attributes to be achieved are maintainability, reusability, performance, and fault proneness. Survey shows that the design patterns support these quality attributes and following table represents some of the quality

Maintainability	Reusability	Performance	Fault Proneness
Abstract-Factory	Adapter	Facade	Singleton
Composite	Command	Command	Template
Decorator	Visitor	State	State
Observer	State		Adapter
Acceptor-Connector	Builder		Observer
	Acceptor-Connector		

Fig. 2 The impact of design pattern in quality attribute in data mining

attributes and their corresponding design patterns used in the data mining and machine learning environment (Fig. 2).

5 Conclusion

Patterns are utility that converts problems in a certain situation into a solution, and are something which is meaningful to all users. There are certain benefits of using design patterns in different components of data mining architecture. From the result analysis, it is clear that by applying various design patterns in different components of data mining architecture, we can improve the quality attributes such as reusability, maintainability, extensibility, adaptability, performance, fault prone-ness. Thus, we can conclude that the impact of software design patterns in data mining architecture is an efficient approach to increase the quality of components.

Acknowledgments This work is supported by the DST Funded Project, (SR/CSI/81/2011) under Cognitive Science Research Initiative in the Department of Computer Science, Amrita School of Arts and Sciences, Amrita Viswa Vidyapeetham University, Kochi.

References

1. Douglas C. Schmidt, "Acceptor-Connector An Object Creational Pattern for Connecting and Initializing Communication Services", An earlier version of this paper appeared in a chapter in the book Pattern Languages of Program Design 3.
2. Boris Delibaši, Kathrin Kirchner, Johannes Ruhland, "A Pattern Based Data Mining Approach", Conference paper. January 2007, DOI:[10.1007/978-3-540-78246-9-39](https://doi.org/10.1007/978-3-540-78246-9-39).source: DBLP.
3. Rodrigo M. L. M. Moreira, Ana C. R. Paiva, Atif Memon, "A Pattern-Based Approach for GUI modeling and Testing", 978-1-4799-2366-3/13/\$31.00 ©2013 IEEE.
4. S. Saira Thabasum, Dr. U. T. Mani Sundar, "A Survey on Software Design Pattern Tools for Pattern Selection and Implementation", International Journal of Computer Science & Communication Networks, Vol 2(4), 496–500.
5. Georges Edouard, "A Software Architecture for Date Mining Environment", KOUAMOU National Advanced School of Engineering, Cameroon.
6. Jasdeep Singh Malik, Prachi Goyal, Mr. Akhilesh K Sharma, "A Comprehensive Approach Towards Data Preprocessing Techniques and Association Rules".
7. Ampatzoglou A, Apostolos K, Elvira Maria A, Antonis G, Fragkiskos C, Ioannis S, "An empirical investigation on the impact of design pattern application on computer game defects." In Proceedings of the 15th International Academic Mind Trek Conference: Envisioning Future Media Environments, pp. 214–221. ACM, 2011.
8. A.V. Krishna Prasad, Dr. S. Ramakrishna, "Software Architectures Design Patterns Mining for Security Engineering", et al./ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1 (5), 2010, 408–413.
9. Francesca Arcelli Fontana, Marco Zaroni, "A tool for design pattern detection and software architecture reconstruction", Information Sciences Volume 181, Issue 7, 1 April 2011, Pages 1306–1324.

10. Gamma E, Helm R, Johnson R and Vlissides J, "Design Patterns. Elements of Reusable Object-Oriented Software".
11. Rudolf Ferenc, Arpad Beszedes, Lajos Fulop, Janos Lele, "Design Pattern Mining Enhanced by Machine Learning".
12. Ichiro Hirata, kenshero mitsulani, Toshiki yamoka, "A study on systematizing GUI design patterns for embedded system products".
13. Prechelt L, Unger B, Tichy W.F, Brossler P, Votta L.G, "A controlled experiment in maintenance comparing design patterns to simpler solutions", IEEE Transactions on Software Engineering, vol. 27, no. 12, pp. 1134–1144, Dec 2001.
14. Cline M, "The pros and cons of adopting and applying design patterns in the real world, Communications of ACM," 39(10), 47–49.
15. Vokac M, "Defect frequency and design patterns: an empirical study of industrial code", IEEE Transactions on Software Engineering, vol. 30, no. 12, pp. 904–917, Dec. 2004.
16. Gatrell M, Counsell, "Design patterns and fault-proneness a study of commercial C# software," Fifth International Conference on Research Challenges in Information Science (RCIS), vol., no., pp. 1–8.
17. Gary P. Moynihan, Abhijit Suki, Daniel J. Fonseca. "An expert system for the selection of software design patterns: Expert System Journal".
18. Afacan T, "State Design Pattern Implementation of a DSP processor: A case study of TMS5416C", 6th IEEE International Symposium on Industrial Embedded Systems (SIES), vol., no., pp. 67–70, 15–17 June 2011.