International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

A Design Pattern Structure Specification to Extract Statistical Report

Mythily M, Radhika Nambiar, Kethsy Prabhavathy, Iwin Thanakumar Joseph, Deepa Kanmani S

Abstract: ***Design patterns are ecological abstract documents which offer acceptable solutions for the continual issues. It allows developers to communicate in a well-known way with the system design. This paper reviewing set of research articles to know the current trend on design pattern's applications. This research also introduces the structure of a new pattern that acts as data analyzer of any software given as input. This pattern acts as centralized data storage of concrete class objects. The instances are analyzed by various functionalities of analyzer in-order to derive results. It helps to solve some of the day to day problems faced by the software designer. This pattern provides services for a group of concrete members to design set of functionalities in order to support data analytics. The structure and its applicability on software are examined, using a case study.***

Index Terms: Design pattern, Application of design patterns, Data analytics

I. INTRODUCTION

Design pattern is a reusable solution or template for the recurring problems in coding [1]. The patterns typically show relationship and interaction between classes or objects. The idea is to speed up the development process by providing tested, proven development paradigm. Design pattern not a finished design which can be converted to code directly. It's a way to solve particular recurring problems in different situations of software design. It speeds up the processing with the help of the proven paradigm. Design pattern helps to prevent the issue that cause a major problem while coding and improves the readability for the coders and are familiar with the structure of patterns. Software system depends on several factors for the quality of the software design patterns. It uses some of the best practices of the object oriented which is experienced by the developers. It tells the way to deal with the problem and when to apply the solution. Some details are included while describing a design pattern such as pattern name, intent, other well-known name, motivation, applicability, structure, participants, collaboration, consequences, implementation, sample code, implementation, known uses, related patterns. Patterns can be classified into three main categories such as creational, structural, and behavioral. Creational pattern concerns about the process of object or class creation, structural patterns deal with the composition of the classes or objects, behavioral patterns characterize the way in which classes or objects interact with each other and disturb the responsibility. There are total 101 design pattern at present out of that 23 are considered. Design pattern have different phases to solve design problems such as finding the appropriate objects, determine object granularity, specifying object interface, specifying object implementations (classes versus inheritance, programing to an interface not an implementation), putting reuse mechanism to work (inheritance versus composition, delegation, inheritance versus parameterized types), real-time run time and compile time structures, designing for changes (toolkits, framework, application programs). We are focusing on different application of design patterns in order to find its current stand on the industry.

II. RELATED WORK

The design pattern improves the package quality by delivering an answer for every revenant issues. Transformation of associate degree application model exploitation each the structural and activity properties of a pattern is represented in terms of the matter and resolution exploitation Query/View/Transformation (QVT).[2] It represents the remodeling associate application model in an exceedingly systematic approach discriminate patterns to enhance quality code. Resolution of every style pattern is outlined in terms of solution specification a tangle specification and a change specification. That the rules of transformation square measure applied supported the mapping between the matter and answer domain. The pattern that results in the answer model with the pattern properties incorporated among them. The specification is enforced on (solution specification, a tangle specification and a change specification) for a few of the patterns like traveler, Abstract manufactory, Observer, Adapter and State patterns. The implementation of transformation rules of QVT (query / read / transformation) is performed for few patterns like traveler, Observer and Adapter and conjointly in an exceedingly conceive to create it on the opposite patterns. Part the implementation

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Mythily M, Department of Computer Science and Engineering, Karunya Institute of Technology, Coimbatore, India.
Radhika Nambiar, Department of Computer Science and Engineering, Karunya Institute of Technology, Coimbatore, India.
Kethsy Prabhvathy A, Department of Computer Science and Engineering, Karunya Institute of Technology, Coimbatore, India.
Iwin Thanakuamr Joseph, Department of Computer Science and Engineering, Karunya Institute of Technology, Coimbatore, India.
Deepa Kanmani S, Department of Computer Science and Engineering, Karunya Institute of Technology, Coimbatore, India.

specification on another pattern like Bridge, Composite, Decorator, Interpreter, and Iterator patterns.

algorithmic rule like layer perennial neural network and call tree are applied for the patterns detection method. It uses Object-oriented metrics for the preparation of

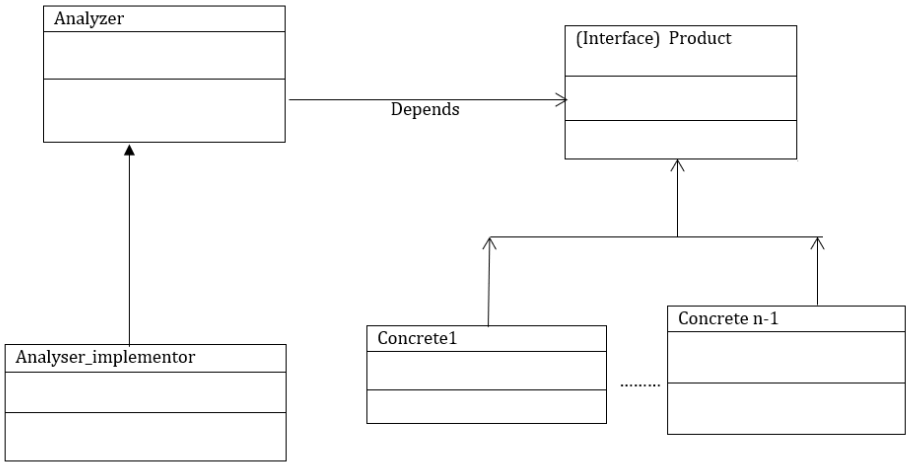

Fig. 1 Structure Diagram of Analyzer Pattern

The selection and classification approach of the design pattern is done using the UML ontology [3] and text categorization. This method includes three phases such as to issues discussion, selection of design pattern by novice organizes and selection of suitable design patterns for the design. The context is proposed with the framework of the three widely used design pattern catalogs and 103 design problems. In other hand supervised learning methods includes, categorization of text which is performed with respect to class labels assigned to documents. Unsupervised learning methods, data, and measures are used for text preprocessing as a prerequisite to text categorization.The text categorization approach has been utilized to every domain to alter organization and choice of style patterns. The gap between the linguistics relationships and between the planning patterns required to be the bridge. By victimization the powerful formula of deep learning that is known as deep belief network that shows the linguistics illustration of documents developed within the style of feature vectors. [4] Additionally enclosed the case study of the text categorization supported the machine-driven system used for classification and choice of package style patterns. The deep belief networks embody adjustment of parameters just like the variety of layers hidden, nodes and iteration which may be employed by a developer to construct an additional versatile feature set. So improves the classifier's performance in terms of organization of style patterns.to judge the effectiveness of the planned approach the 3 wide used style pattern classifiers like call tree, naïve Bayes and support vector machine. Thus, improve the performance of classifiers to arrange the planning patterns. The techniques used for the popularity of code style pattern is machine learning and conjointly used in code development. It's typically desired to spot additionally as knowledge style pattern spot by playacting reverse engineering within the ASCII text file since it improves documentation and maintainability of the ASCII text file. For the popularity, a coaching dataset is developed supported code metrics and machine learning metrics primarily based dataset [5]. The popularity of the planning pattern is performed by victimization the abstract industrial plant and adapter design patterns. These embrace techniques for process datasets that yields higher accuracy by minimizing the quantity of candidate patterns. The most advantage of victimization object-oriented metrics and machine learning techniques is to spot the proper reasonably roles for the pattern participants and develop learning models, that improve quality parameters for the popularity of style.The objective is to evaluate the design patterns to assure in the software quality in terms of design and quality attributes. The impact of patterns should be given adequate attention. [6] Performed number of studies to derive quality of software through design patterns.In [7], it includes model derivation and analysis used for the investigation of model transformation. The quantitative chemical analysis is required to remodel the performance model generation and analysis. This space of the model transformation intent to take advantage of the information and import each reasoning and methodologies in the software system performance engineering. The investigation extracts data from extra-functional properties at the field of study level. The system style may be improved by the standard of internal attributes by applying the proper application. The software system maintenance and existence of up thus far documentation could be a crucial a part of system style. Therefore the invention of unknown style pattern instances will facilitate to boost the standard of documentation. Typically the straightforward strategies will offer in exact results because of the imprecise nature of the pattern's structural description. The incorporation of machine learning strategies so as to refine the results of the structure-based approaches is projected by [8].

Retrieval Number: J95080881019/19©BEIESP
DOI: 10.35940/ijitee.J9508.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

This experiment shows that the model obtained may filtrate fifty one of the fifty nine false hits of the Adapter Object style pattern (out of a complete of eighty four hits) and thirty three of the thirty five false hits of the Strategy pattern (out of a complete of forty two hits). It shows that this approach has the potential to reinforce current style pattern mining strategies any to be created to develop the system.The main motive is to unravel the complicated issues among a given context in software system style victimization style pattern. The method of detection of the look pattern instances from ASCII text file is that the major role in learning the big and sophisticated systems. Detection of style pattern isn't forever an uncomplicated task. The graph theories have divided the detection method into 2 ordered phases [9]. The primary part is regarding each the linguistics and also the syntax of the structural signature of patterns. During this part the patterns are detected and reworked into linguistics graphs. The system graph is broken into system graphs. Within the second part final matches are obtained. Thus, this describes each the structural and behavioral signatures of the patterns and finds the precise UML category diagram from the ASCII text file. Linguistics analysis is employed part within the structural part. So as to separate the structural and behavioral pattern.This [10], reviews the systematic approaches that perform transformation of design patterns. The model transformation in design patterns uses quite widespread and it's often asymmetric. Recent trends motivate the usage of patterns in transformation as well as to introduce new patterns to achieve transformation.Hamid projected [11], associate approach to secure the systems that area unit addicted to patterns and models to specific applications at intervals a selected domain. It focuses on the problems associated with code engineering victimization style employ. This could cut back the value of engineering system that allows the protection problems that area unit addressed in associate early system development method together with the technical details. The event of application victimization the pattern-based technique method associated reusing the present patterns in associate acceptable means for the targeted development setting is an important policy of pattern reusability. The growing quality is handled by the approach of system vogue and most of the favored models use main artifacts for construction and maintenance. The model-driven approach methodology is said to a pattern-based approach to support the event of secure code. [11] Is based on the met modeling and model transformation techniques to stipulate the patterns whole completely different in many levels of abstraction and generate different illustration as per the domain issues. The implications of security and use practices on total system and software quality unit of measurement assessed by usability testing at intervals the trade.

III ANALYZER PATTERN

A pattern developed to get information from the concrete class objects and produce different dimension of information in order to perform data analytics and it is named as Analyzer pattern. Querying the information from different classes without affecting their behavior is considered as major benefit of this pattern. It is a centralized approach to add objects of concrete classes.

A. Benefits of Pattern

- ❖ Analyzer is a Reusable structure with set of functionalities in order to provide service for the group of concrete members.
- ❖ Concrete classes are independent of analyzer that focuses on domain aspects.
- ❖ Centralized data storage at analyzer.
- ❖ Easy extend ability of analyzer.

B. Intent

- ❖ Analyzer is defined as a class for creating an object and adding into the array list and concrete classes focus on the other aspects.
- ❖ Product class is a single point of contact to all concrete classes in order to achieve dynamic method dispatching.
- ❖ Contains concrete classes which are related to each other through the objects.

C. Structure Diagram

The structure diagram of the proposed analyzer patter is shown in fig 1. Each component of the structure is described as follows,

I. Analyzer

It stores collection of objects of various concrete classes as per the module and further it is inherited by the concrete analyzer in order to analysis various aspects of the above store objects based on the user requirements. It is a ready-made solution that can be used for many applications based on their requirements. However, the structure remains the same. Due to this the main objective of design patterns will be achieved in order to sort out a particular issue of the user as a case study we have taken a set of student objects, subjects that are register by the students and as well as internal exam. This collection of objects has interconnection between them. And the values that are generated using these objects can be analyzed with respect to various aspects. This solution supports to monitor the performance of the student and the result of each for each subject. By the help of the analyzer any result analysis that is to be derived can be implemented as member functions and get the required results.

II. CONCRETE CLASSES

That has an implementation for all of its methods. They cannot have any unimplemented methods. It can also extend an abstract class or implement an interface as long as it implements all their methods.

They are the basic class used to observe the content from the real time environment.

III. ANALYZER_IMPLEMENTOR

It gets the data stored in the analyzer and it also specifies its own kind of member functions which supports to analyses the group of objects. The main purpose of this class is to have set of objects of all the concreter classes.

Table I: Use Case Description Table for Case Study

Name	Result analysis
Description	This solution supports to monitor the performance of the student. And the result of each and every subject
Actors	Student, Subject, Internals
Organizational benefits	The solution will help the organization to get the result of the students and details of the number of students pass and fail for each subject
Per-condition	• Student object • Subject object • Internal object
Post-condition	• Internal class should contain the subject object. • Student should have all internal marks and subjects.
Main course	Concrete analyzer includes some functions such as: • Process • Number of pass students • Number of fail students.

III. CASE STUDY

Student's internal mark result analysis scenario is taken as the case study. A user description of the case study is given in table 1. The classes Internal, Student and Subjects are the concrete classes to support internal exam conduction. The interface called internal_assessment supports all the concrete Class in order to make a perform dynamic method dispatching. A set of functionalities represented in table. 2 provided the usage of analyzer patter in the given case study. The case study is represented in the structure of analyzer pattern in fig 2. The individual domain classes are discussed as follows,

Analyzer class -It is a module which contains objects of all concrete classes which are added into array list. The analyzer design pattern is used this case study which helps in analyzing the student result and keep a track on the progress.

Table II: Sample Analytics by the Analyzer

Sample Functionalities in case Study	**Concrete class dependency**
No of students passed in a subject	Subject, Internal
Mean and Median of a particular subject result	Internal
A particular student's marks for an internal	Internal
average of a student	Student
Average of a subject	Subject
Identifying name of faculty for a subject	Subject

Subject - It is a basic concrete class used to observe the content from the real time environment. It includes subject name, batch faculty name and number of students registered as data members.

Student-It is a basic concrete class which includes student register number, year, semester, marks for each subject as data members. It observes the content from the real time environment.

Analyzer_implementor - It inherits the analyzer class and contains some functions which are performed by using the array list in the analyzer class. It includes functions like process, display, number of pass students as per subjects and number of fail students as per subjects.

Internal -It is a basic concrete class used for observe the real time content from the environment. it includes internal name, subject name, batch, marks for the subjects as the data members.

Internal_assessment - It's an interface module between all

not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

the all the three concrete classes which implements
the interface and overrides the functions which are present in interface.

IV. CONCLUSION

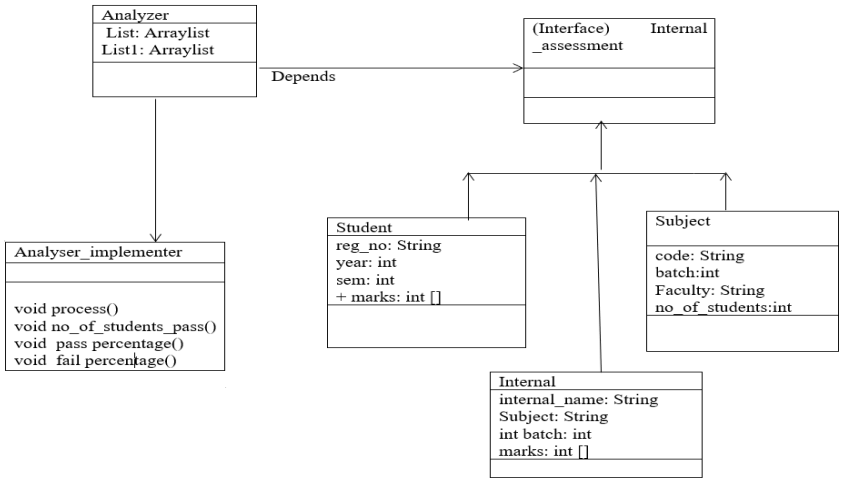

Fig 2: Student Internal result analysis report in analyzer pattern

The design pattern used as a tool to solve the issues of concrete classes' data analysis by implementing analyzer pattern. It is a common solution for any application that requires object analysis over certain criteria. This analyzer is an extensible feature where any new queries on data can be introduced. This analyzer is used for the communication between the concrete classes. Object initializations of all concrete classes are under the control of analyzer in order to maintain the integrity on the result. The proposed pattern is analyzed by implementing a result analysis report of students' internal marks. This case study resultant in a better performance with respect to cohesion and coupling.

REFERENCES

1 B. E. Lorber, "Un Nid De Carton De Dendrolasiuis Fuliginosus (Hym., Formicidae, Formicinae)," Entomologiste, Vol. 36, Pp. 135–139, 1980.
2 D. K. Kim, L. Lu, And B. Lee, "Design Pattern-Based Model Transformation Supported By Qvt," J. Syst. Softw., Vol. 125, Pp. 289–308, 2017.
3 S. Hussain, J. Keung, M. K. Sohail, A. A. Khan, And M. Ilahi, "Automated Framework For Classification And Selection Of Software Design Patterns," Appl. Soft Comput. J., Vol. 75, Pp. 1–20, 2019.
4 S. Hussain Et Al., "Implications Of Deep Learning For The Automation Of Design Patterns Organization," J. Parallel Distrib. Comput., Vol. 117, Pp. 256–266, 2018.
5 A. K. Dwivedi, A. Tirkey, R. B. Ray, And S. K. Rath, "Software Design Pattern Recognition Using Machine Learning Techniques," Ieee Reg. 10 Annu. Int. Conf. Proceedings/Tencon, Pp. 222–227, 2017.
6 M. N. Riaz, "Impact Of Software Design Patterns On The Quality Of Software: A Comparative Study," 2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. Icomet 2018 - Proc., Vol. 2018–Janua, No. 4, Pp. 1–6, 2018.
7 A. Di Marco And R. Mirandola, "Model Transformation In Software Performance Engineering," Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Vol. 4214 Lncs, Pp. 95–110, 2006.
8 R. Ferenc, Á. Beszédes, L. Fülöp, And J. Lele, "Design Pattern Mining Enhanced By Machine Learning," Ieee Int. Conf. Softw. Maintenance, Icsm, Vol. 2005, Pp. 295–304, 2005.
9 B. Bafandeh Mayvan And A. Rasoolzadegan, "Design Pattern Detection Based On The Graph Theory," Knowledge-Based Syst., Vol. 120, Pp. 211–225, 2017.
10 K. Lano, S. Kolahdouz-Rahimi, S. Yassipour-Tehrani, And M. Sharbaf, "A Survey Of Model Transformation Design Patterns In Practice," J. Syst. Softw., Vol. 140, Pp. 48–73, 201
11 B. Hamid And D. Weber, "Engineering Secure Systems:Models, Patterns And Empirical Validation," Comput. Secur., Vol. 77, Pp. 315–348, 2018.

AUTHOR PROFILE

M. Mythily has completed her bachelor and master degree with a specialization of computer science and engineering from Avinashilingam Deemed University and Government college of Technology, Coimbatore, Tamilnadu, India respectively. Currently, She is pursuing her Ph.D degree with a specialization of software engineering at Anna University, Coimbatore. At present working as an Assistant Professor at Karunya Institute of Technology and Sciences. As a wholesome she gained a decade experience in industry as well as in teaching. Her area of interest includes Software engineering, Machine learning and Problem solving techniques

A.Radhika has completed her bachelor degree with a computer science and engineering from Karunya Institute of Technology and Sciences. Currently she is pursuing her Master in computer science and engineering from Karunya Institute of Technology and Sciences

A. Kethsy Prabavathy received her Bachelor of Engineering degree in 2004, Master of Engineering in 2008 and Ph.D (Computer Science) in 2018. She is currently working as an Assistant Professor in the department of Computer Sciences Technology in Karunya University, Coimbatore. Her research interest is mainly based on Image processing, Image segmentation and Video segmentation.

S. Iwin Thanakumar Joseph is an assistant professor in the department of computer science and engineering, Karunya Institute of Technology and Sciences, since 2008. He received his Masters degree M.E in Computer Science Engineering, Karunya University, Coimbatore, 2008 He is pursuing his Ph.D in the department of Computer Science and Engineering department, Annamalai University, His research interest includes Artificial Intelligence, machine learning, image processing, Internet of Things.

S. Deepa Kanmani currently pursuing PhD in the stream of Computer Science and Engineering at Karunya Institute of Technology and Sciences. Areas of interest are Datamining, Big dta analytics.