Software Design Pattern & Mining Approach – A Novel Query Search Analysis to Identify Relevant Information

Kaluva Manasa
(Ph.D) CSE
K.L University

ABSTRACT

Design pattern is a reusable module to improve the problem and defines solution among all those methods techniques & algorithms to choose the best one then applies for our requirement. When designing a system or end product usually need to control how an object is used and prevent users from making copies of it or creating new instances. Design pattern are used in object oriented programming language, Dot Net, Java. Design pattern is simple research but it is innovative, to achieve our analysis for particular query in search engine, we are using design pattern to identify the solution in development. In this paper define a problem & solution using of design pattern our aim is to detect unwanted information in particular analysis and provide the relevant information, to retrieve the information from databases our solution may use the mining algorithm and design pattern in the development of our end product "Novel Design Pattern Mining". This work improves the mining process and reduces the time and effort for developers.

Keywords

Design Pattern, Search Engine, Mining Technique, Object Oriented Language.

1. INTRODUCTION

Design patterns are optimized we can reuse them in programming problems that encounter every day. Pattern is not a class that can simply plug into our system it is a template that has to be implemented in the correct situation. A good design pattern should be implemented in most of the languages depending on the capabilities of the language. By principle design patterns to solve programming problems, before going to implement software programmers have encountered problems in programming languages design pattern is the solution to remedy them. Design pattern represent solutions to problems that arise when developing software within a particular context. Pattern captures the static and dynamic structure collaboration among key participants in software designs particularly useful for articulating how to resolve non-functional problems.

Pattern = Problem/Solution pair in a context.

Merge two classes which do two different things based on the situation these are used by the existing system in different places making it difficult to remove these two classes and change the existing code. To add this changing the existing code requires that also need to test any changed code, with these edits in a system relies on different components almost always introduce new bugs. Instead of implement a variation of the strategy pattern. Strategy pattern is a functional design pattern allows to decide which context of action a program should take place inside two classes at runtime, it defines algorithms in one family encapsulate each one and make them interchangeable and vary independently from clients that use it. Imagine currently developing a class which either update or create a new user record needs the same inputs but depending on a given situation it has to use different functions when updating strategy pattern involves encapsulating all algorithms inside another class but in adapter another class would be wasteful. Adapter pattern is s structural design pattern allows repurposing a class with a different interface allowing it to be used by a system which uses different calling [1] methods. Convert the interface of a class into another interface clients expect, classes couldn't work together because of incompatible interfaces.

Designing a software system is hard high quality reusable system is even harder design patterns may help to design large high quality systems in forward engineering. Pattern related knowledge is generally not available in software system. Understanding large software system is difficult many legacy systems lack design documents with evolution of software systems design documents may not be consistent with their source code. Reverse engineering [2] design information can recover that previous decisions and assist the understanding of the systems and improve the system with higher quality. Many existing techniques tools can help mining software design patterns from system source code.

2. RELATED WORK

In a given context, Pattern is predefined solution for a problem. We have three keywords in the definition Context, Problem, Solution. Patterns are to enhance our system reusability and repeatability which returns Cost efficiency and Agility. Building architecture and design such as pyramids roman collegium used the patterns in the implementation.
Conceptual Patterns: Conceptual model of the application that evolves the system understandable by all parties concerned. Usually consists of a set of related concepts and terms of the application domain comprising the different viewpoints of the various groups involved in the software development process. Conceptual pattern comprise guideline for perceiving interpreting and changing the world. Link between the real world and the conceptual model can be strengthened. These metaphors as understandable "mental pictures" are supportive when taking the step from the situation at hand to the design of the future system. Conceptual patterns should be based on metaphors rooted in the application domain don't serve a general purpose, always have to find the balance between too specialized patterns and contexts. If patterns are too abstract it is general to really guide analysis and design. Very specific pattern might not be used beyond a single project and therefore will not become part of everyday practice and a development culture. The distinction of tools and materials pattern on the right level of abstraction for workplace computing but not used for real time software and [3] conceptual pattern should be geared towards a restricted application domain.

Programming Patterns: Developing a software system our analysis shows the application domain with the software design model. Technical artifacts which run as a formalism on a computer and utilized as an application supporting users. Programming languages provide the notation. A programming pattern is a pattern whose form is described by means of programming language constructs. We use these patterns to implement a software design based on programming patterns vary among programming cultures.

```
For (i=0; i<max; i++)
```

We found above syntax in every C program.

It is important to realize that even on low technical level there is something like programming cultures and styles.

3. PATTERNS

3.1. Pattern Format

The two types of pattern are Alexandrian form and GoF format.

Alexandrian form describes the pattern description format adopted from work of Alexander Christopher is referred to as canonical form. GoF format is the format describing patterns proposed by Gamma et al.

Name: *Must have the meaningful name*. Good pattern names form a vocabulary for discussing conceptual abstractions.

Problem: *Statement of the problem the pattern is trying to solve.* Problem contains the statement of the pattern to solve a general idea behind a description of the problem in the pattern that the problem description should give a designer knowledge of the pattern solves which enables the designers to know when to apply given pattern.

Context: *A situation giving rise to a problem.* Checks the preconditions under which the problem and its solution to recur for which the solution is desirable. A pattern solves a problem in a given context and usually it might not make sense in the other context.

Forces: represents a concrete scenario provides the use of pattern in certain context to solve a problem. Forces make clear intricacies of a problem[2]. A good pattern description should encapsulate all the forces upon which it might have an impact.

Solution: *Proven solution to the problem.* Solution describes not only static structure but also dynamic behavior of the pattern static structure represents the form and organization of the behavioral dynamics is what makes the pattern alive.

Example: *Sample of the application of the pattern.* Existence of reflection of a good pattern should be one or more sample applications which illustrate the specific initial context.

Rationale: Rationale provides insight into the deep structure key mechanisms where the pattern comes from why it works and expert use it.

GoF format: It is classified into number of sections, pattern in GoF [3] format must have a Name additionally Classification of the pattern should be stated refers to the classification according to Gamma et al. sometimes pattern may have more than one commonly used name. Intent describes the design issue patterns is addressing thus reflecting what a problem and how the class and object structures in the pattern solve this problem. Applicability describes situations in which the design pattern can be applied as well as poor designs the pattern is addressing with hint how to recognize such situations.

3.2. Needs for Pattern Design

Pattern needs to do the process of problem solving and program design, the majority of texts are driven by the syntactic details of a specific language rather than by general methods for solving problems and designing programs. Patterns capture established practices that remain obscure in the broad practice of a given domain, many patterns have their work of early adaptors of a new technology or the first architects of a system. Many of patterns attack problems in subtle ways which makes it difficult to cast them in the framework of the predominate constructs of the system. For example in object oriented languages first need to learn classes, functions and objects or in terms of object-oriented design principles such as inheritance and polymorphism that lead to good class partitioning. Educational modeling language an XML based language for specifying learning activities in domains and cornerstone in the specification of IMS learning design. Significance lies integrating learning design principles and activities into machine-readable international standards for eventual adoption by various e-learning tools. Design patterns have been proposed as a common language between educators and technologists to help educators develop pedagogically sound learning design scenarios.

3.3. Artifact for Design Pattern

Pattern is reusable can be any information which a developer may need in the process of developing object.

Code fragment come in a form of source code or various charts

Logical program structures such as modules interfaces of data structures.

Functional structure specification of functions and their collections.

Knowledge of development process in a life cycle models.

Environment level information experimental data.

Controlled collection of reuse constitutes a library must contain not only reusable components but are also expected to provide certain types of services to their users example storage searching inspecting and retrieval of artifacts from different application domains and of varying granularity and abstraction loading linking and invoking of stored artifacts relationship. The major problems in the utilization of such reuse libraries are in determining appropriate artifact classification schemes. To avoid problems with reuse libraries the use of specialized domain specific language was introduced as an alternative.

Certain classes of software artifact identified as eminently suitable to become reuse.

Expressive general utility and of adequate level of abstraction so that they could be used in many different contexts and be applicable to variety of problem areas.

Definite constructed the capabilities and limitations are easily identifiable interfaces required resources external dependencies and operational environment are specified requirements are well defined.

Additive is possible to seamlessly compose existing artifacts into new products or other reusable components without the need for massive software modification.

Formal reusable artifacts should at least at some level of abstraction be described using a formal notation such an approach provides means to formally verify an artifact correctness it enables to predict violation of integrity constraints during artifact composition.

Machine representable artifacts which can be described in terms of computationally determined attribute values which can easily be decomposed into machine representable parts which can be accessed analyzed manipulated and possibly modified by computer-based.

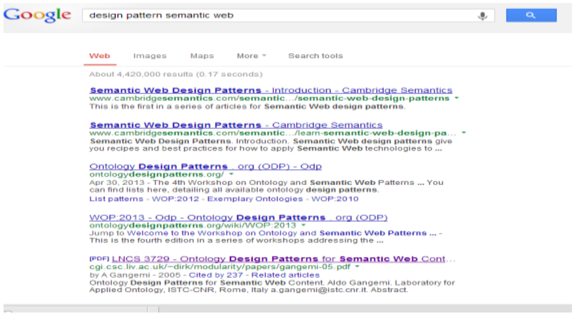
4. PROBLEM DEFINITION

Semantic web are ontologies presents often hand-crafted, challenge is to learn ontologies instance of their concepts in a semi-automatic way by using web mining techniques such as association rules. Web mining is the application of data mining technique to the content structure and usage of web resource, primary web resource being mined is a record of the requests made by visitors to a website most often collected in a web server log.

Mining the semantic web graph is a complex problem, it suite of standard meant to enable the creation of a machine readable web of data and web data could then be used by intelligent agents to share on global scale. World Wide Web doubles in size approximately every 8 months 20 million content areas in the web. 85% of users use search engines to locate and buy goods to research many decisions. However search engines are currently lacking in comprehensive and timeliness and do not index sites equally.

Research in web mining is ultimate goal of developing [4] computational approaches for monitoring public opinion in regions of conflict, indicators, and social media correlating these risk signals with commonly accepted quantitative assessments. Serious concern to the international community country risks has traditionally been assessed by monitoring economic indicators. To retrieve the relevant content for particular query our research framework presents design pattern that offer security at the architectural level have been proposed in analysis.

When we search for a particular subject text or information, the result of query analysis shows different related information which we have given query in search engine. Design pattern is a method to retrieve exact information related to our query, pattern is relevant information known as function module.



About 4,420,000 results (0.17 seconds)

Semantic Web Design Patterns - Introduction - Cambridge Semantics
www.cambridge**semantics**.com/**semantic**.../**semantic-web-design-patterns**
This is the first in a series of articles for **Semantic Web design patterns**.

Semantic Web Design Patterns - Cambridge Semantics
www.cambridge**semantics**.com/**semantic**.../learn-**semantic-web-design-pa**...
Semantic Web Design Patterns. Introduction. **Semantic Web design patterns** give you recipes and best practices for how to apply **Semantic Web** technologies to ...

Ontology **Design Patterns** . org (ODP) - Odp
ontology**designpatterns**.org/
Apr 30, 2013 - The 4th Workshop on Ontology and **Semantic Web** Patterns ... You can find lists here, detailing all available ontology **design patterns**.
List patterns - WOP:2012 - Exemplary Ontologies - WOP:2010

WOP:2013 - Odp - Ontology **Design Patterns** . org (ODP)
ontology**designpatterns**.org/wiki/WOP:2013
Jump to Welcome to the Workshop on Ontology and **Semantic Web Patterns** ... - This is the fourth edition in a series of workshops addressing the ...

[PDF] LNCS 3729 - Ontology **Design Patterns** for **Semantic Web** Cont...
cgi.csc.liv.ac.uk/~dirk/modularity/papers/gangemi-05.pdf
by A Gangemi - 2005 - Cited by 237 - Related articles
Ontology **Design Patterns** for **Semantic Web** Content. Aldo Gangemi. Laboratory for Applied Ontology, ISTC-CNR, Rome, Italy a.gangemi@istc.cnr.it. Abstract.

Figure 1 Search engine result example which doesn't match to our query.

5. USING DESIGN PATTERN TO DEFINE SOLUTION

Retrieving the matching of some user query against a set of information records could be any type of unstructured text like newspaper articles business related issues education photography several numbers of paragraphs in a manual. Retrieving is sometimes referred to as text retrieval branch where the information is stored primarily in the form of text. We can store data in some CD-ROM personal computer now days in any cyber using the search engines.

Retrieval system find the information to given criteria by matching text records against user queries as opposed to expert systems that query answer by inferring over a logical knowledge database. To find relevant documents to user queries search evaluates exact and sort them according to relevance using some algorithms such as PageRank Boolean systems. An image retrieval system is a computer system for browsing retrieving images from complex database of digital images.

Design pattern addressing concern related to high level strategies describing to exploit application characterization on a computing platform and key role identification, includes strategies in executing streams of tasks and building blocks to support synchronization. Design patterns have also been made to codify design patterns in specific domain include use of previous design patterns as well as domain were combined into different categories

For example if we take payment online details.

The requirements for the solution say that it must accept the request from the fulfillment system and verify that the payment is either debit card verification or credit card verification from figure below.

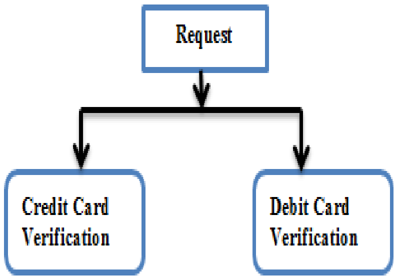


Figure 2 Classify the Design Pattern.

Taking these requirements into consideration the structure lends itself to the design pattern. The solution state that an external passes an account number the total amount that must be verified and a payment mode to the solution based on the requirement. Important tool in the analysis and design of software end product development pattern give the complete architecture the ability to conceptualize a solution at several levels to improve the quality. All solution architecture and design map the functional requirements by implementing patterns in the design of domain specific framework to provide consistency throughout the go-live product.

6. DESIGNING A DESIGN PATTERN

We describe patterns using a consistent format each pattern is classified into sections according to the following template. It is uniform structure to the information making design patterns easier to learn compare and use.

Pattern Name and Classification: The pattern name is the essence of the pattern succinctly a good name is vital role. It is vocabulary for design pattern.

Intent: Short statement that answers the following questions what does the design pattern do what is its rationale and intent.

Also Known As: if any well-known name for the pattern.

Motivation: Scenario that illustrates a design problem and how the class and object structures in the pattern solve the problem.

Applicability: What are the situations in which the design pattern can be applied, how to recognize these situations.

Structure: Graphical representation of the classes in the pattern using a notation based on the object modeling technique.

Participants: The classes or objects participating in the design pattern and their responsibilities.

Collaborations: How the participants collaborate to carry out their responsibilities.

Consequences: How does the pattern support objectives, tradeoffs and results of using the pattern. What aspect of system structure does it let vary independently.

Implementation: what pitfalls hints or techniques should you be aware of when implementing the pattern.

Sample Code: Code fragments that illustrate how you might implement the pattern.

Known Uses: Examples of the pattern found in real systems. We include at least two examples from different domains.

Related Patterns: What design patterns are closely related to this one.

7. EVALUATION

In our work design pattern web text is most innovative role to identify relevant text or information. Actually if we search for a query in Google will get so many pages related to our query we cannot decide which one is better to choose the particular information all the information contain in different pages would be same.

Design pattern is to apply different phases of software development

Pattern approaches are not bound to some specific part of software development pattern documented to range from the concern requirement engineering to process management or software architecture.

Different levels of relationships

Some approaches present standalone patterns while other present patterns that relate to each other to make up language or a system

More or less domain-specific dependency

Some patterns describe problems that are more domains specific for example a pattern that describes how to use a database is far more specific than a pattern that describes how to create a complex object.

There are numerous forms of describing a pattern of course this depends on which type of problem the pattern address. A pattern describes how to organize people to manage software will different from a pattern that describes how to arrange three layered architecture.

Object oriented design patterns include program to an interface not an implementation and favor object composition over class inheritance. Advantages of clients remain unaware of the specific types of objects they use as long as the object adheres to the interface clients remain unaware of the classes that implement these objects. Use of an interface also leads to dynamic binding and polymorphism.

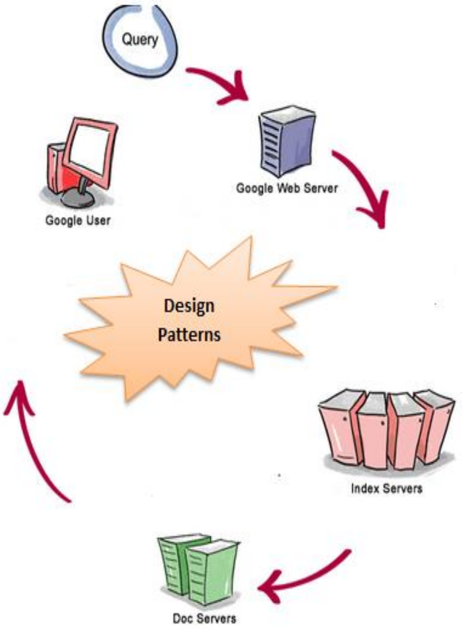


Figure 2 is Proposed Architecture

Step 1: Enter the query in search engine to retrieve the text.

Step 2: Apply design pattern to the analysis of text retrieve process.

Step 3: Preprocessing the relevant data from database.

Step 4: Transforms into the pattern and provides the end user particular information.

8. CONCLUISON

Design pattern is a reusable technique applied in the development process before the Design phase which improves the problem and provides the solution among all the problems. Using this design pattern in any application like Java Dotnet or any other programming languages detect noisy and integrates the different models, finally our information patterns satisfies the need of client inputs. Our contribution provides the work on relevant information using query search engine system which is known as “Novel Design Pattern Mining “ user enters the query in search engine, entire databases contains index search, document search the query processes the entire database finally provide the result this process is implemented using mining technique. Design phase our system uses the design patterns, the end product is developed in Java. Future work extends to survey all the design pattern types then will give comparison on each pattern and at present design pattern only used in object oriented language, Dotnet will survey on other programming languages uses of design patterns.

9. REFERENCE

[1] Admodisastro N.S Palaniappam 2002 Tool for the gamma Design Patterns. Malaysian Journal of computer Science 15 (2) 94-101**.**

[2] C. Alexander. The Timeless Way of Building. Oxford University Press, 1979.

[3] B. Appleton. Patterns and software: Essential concepts and terminology. http://www.enteract.com/˜braddapp, 2000.

[4] Tong, Y., & Chen, H. (2008). Semantic graph mining for biomedical network analysis. In Proceedings of the WWW Workshop on Semantic Web for Health Care and life Science.

[5] Candy P 2004 linking thinking Self-directed learning In Steeples & C.Jones (Eds) Networked learning: Perspectives and issues (pp. 293-308) London Springer.

[6] Y J.L Qin L.Chang 2009 Keyword search in databases. In synthesis Lectures on Data Management Morgan & Claypool Publishers.

[7] Manolescu, Dragos; Markus Voelter and James Noble (2006). Pattern Languages of Program Design 5. Addison-Wesley.ISBN 0-321-32194-4.

AUTHOR’S PROFILE

Kaluva Manasa pursuing Ph.D Computer Science Engineering from KL University MS from McNeese State University. Currently working as Asst Prof in Avanthi Group of Institute & having seven years of Academic experience and guided many UG & PG students. Areas of interests include Data Mining Networking Software Engineering & Software Design Patterns.