International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217369

An Efficient Comparison between Structured Analysis and Object-Oriented Analysis

Maulik V. Dhamecha
Computer Engineering Department, VVP Engineering College, Rajkot, Gujarat, India

Article Info
Volume 7, Issue 1
Page Number: 290-295

Publication Issue :
January-February-2021

Article History
Received : 05 Jan 2021
Accepted : 20 Feb 2021
Published : 25 Feb 2021

ABSTRACT

if you want to find out that which is the best suitable analysis then you have to find outs all the merits and demerits of that both analysis methods. The aspect of this paper is to describe combine both the traditional (structured) and the object-oriented approach, certain methodology for information systems development. Despite the fact that objects-oriented paradigm is actually widely adopted for software analysis, design, and implementation, there are still a large number of companies that continue to utilize the structured approach to develop software analysis and design. The empirical study that we present considers both an Uncontrolled and a controlled experiment with Master students. with the existing structured approach of developing a system there is significant way to know how appropriate the OO topics. So, in recent time, Objects-orientation analysis is largely acceptable subject. This paper discourses some basics about these two design paradigms.

Keywords : Structured, Object-Oriented, Objects Oriented

I. INTRODUCTION

Now a days, Computers are rapidly becoming an integral part of nearly every engineered product, as well as controlling the manufacturing process for products: Computers control consumer products, commercial aircraft, nuclear power plants, medical devices, weapon systems, aerospace systems, automobiles, public transportation systems, and so on. Virtually nothing is engineered and manufactured in the U.S. today without computers affecting the design, manufacturing and operation. Not only do products use computers to operate better or cheaper---``smart'' automobiles and appliances are examples---but complex systems are incorporating designs that cannot be operated without computers---for example, unstable aircraft and space vehicles that cannot be operated successfully by humans alone.

At the same time that computers are becoming indispensable in controlling complex engineered systems, quality and confidence issues are increasing in importance. We are hearing more and more about failures due to computers: Software errors have resulted in loss of life, destruction of property, failure

of businesses, and environmental harm. Computers now have the potential for destabilizing our financial system. Some large government-financed projects are in trouble or have been canceled because of difficulty in assuring the quality of the software.

One of the reasons for the problems is that these systems require that standard engineering techniques be extended to deal with new levels of complexity, new types of failure modes, and new types of problems arising in the interactions between components. Computers exacerbate engineering problems by allowing levels of complexity and coupling with more integrated, multi-loop control in systems containing large numbers of dynamically interacting components. We are attempting to build systems where the interactions between components cannot be thoroughly planned, understood, anticipated, or guarded against. The fundamental problem is intellectual unmanageability: Increased complexity and coupling make it difficult for the designers to consider all the potential system states or for operators to handle all normal and abnormal situations and disturbances safely and effectively. The failures in these systems are arising in the interactions between components. While we train engineers to be experts in individual fields, these complex heterogeneous systems (composed of electromechanical, digital, and human components) require knowledge and techniques that span engineering disciplines.

II. FUNDAMENTAL CONCEPT

Structured Projects

Many of the current versions of the System Development Lifecycle (SDLC) are some modifications of the traditional waterfall scheme. From a structured waterfall model, through incremental, spiral, next generation, win-win, V and W and WEIT models, it is possible to track joint leads to improve cost/performance management and user engagement. In addition, many special SDLCs and some SDLCs, such as 6 Sigma, have been added. Each new SDLC provides system analysts with a roadmap for improvement to develop and manage information systems projects. In many cases, these projects are very challenging and require a lot of resources. There are notorious and well-documented failures, but in general, a large number of challenging projects have been implemented and will continue to be successfully implemented.

While comparing or presenting these SDLC models, regardless of name, each individual model, can be boiled down to four main parts: delivery, design discovery and development. For a project proposal, the specific roadmaps are almost the same: 1) know exactly where you need to go, 2) challenge before you know where you are 3) by what means you have to move from here to the place. for taller structure projects, this process has been tried and tested to works great. Such projects can be called "engineering", such as from scratch to building a parking lot. cleaning the floor before installing the drain, etc. Before actual work begins, the final configuration is known and approved. Just follow the instructions on the SDLC sequences.

Although without serious discussion textbooks of current systems often offer a course SDLC model, sometimes no more than numerous paragraphs, examining the class order reveals the traditional linear incremental SDLC model. As the structures can be misleading for new projects based on Internet that are difficult to identify easily. As an example, designing the output usually falls in Chapter 7, although the user may need to experience the output very early in the process and provide feedback. This is why there are so many different (linear) SDLC models to choose from and why more serious discussions about SDLC are needed. To do this, an object-oriented SDLC has been added.

Unstructured Projects

Many unstructured ("mysterious") projects, those are information systems projects and not long or exaggerated. The journey begins before the final destination is fully determined. You only know where to go. These projects can be called "social". The design suggests modules, but the individual variants look different, although there are specific features. defines the first order with its attributes into its journey and then combines the attributes as it evolves. Only at the end of a certain stage does a valid (if not yet clear) form of the project appear. This process is new, below. This is where the SDLC object finds a new design-oriented systems analysis and design (OSAD) format that can handle not only structural but also disorganized designs.

OSSD has four main steps and follows a sequence like its structured complements, The sequences are not new: research (startup), design (expansion), development (construction) and delivery (transformation). As per the need, by splitting these steps can be changed. For educational purposes, it is useful to start with general terms and then explain the difference as they once did with "strings" and "tuples". In fact, there is something new here.

However, the sequence itself is now completely different. It is not developed using traditional measurement tools and also not a linear. In fact, there is still not a fix sequence of what an OSD sequence would be presented and how it will be demonstrated. It's not really helpful advice on how best to present a roadmap, especially for students who have already experienced traditional SDLC. This is the work of the following.

III. Similarity between Object-Oriented Analysis and Structured Analysis

If we want to learn difference between Object-Oriented Analysis and Structured Analysis then we want to study first of all that what is the similarity between Object-Oriented Analysis and Structured Analysis

Since at least the mid-1980s, system parts that respond autonomously to exterior or temporary incentives for system analysis that are preferred technique for constituting parts of a system. from where the preliminary system models of the system can be obtained, The results of the incident analysis are displayed in the event table. It categorizes the main set of incident analysis processes in structural approach. In object-based analysis, each event leads to a case study of the underlying use.

Within the application domain, the Entity relation diagram provides a conceptual model for relationships and organizations into the structural analysis model. This is done by simplifying the data stores in a Data Flow Diagram (DFD) suite. Whatever role plays an Entity relation diagram same role will played by domain model into the object-based analysis. For segment names and graphic assemblies, that two models are almost different. correlation in UML clearly show the relationship in both models. However, foreign keys are required to implement the relational model; These external features are redundant in the domain model. In the detailed entity relationship diagram, the UML model includes the specialization and chain of generalization.

Every time many Analyses explains “what users need” but not “how to meet those needs”. The best practices in systems analysis have always addressed the needs of users in a way that by including details of applied technology does not discriminate in a

design solution. In Analytical process of structured data Analysis captures user needs in a set of basic data flow diagrams, complemented by a system dictionary that contains details of basic data flows, basic data stores, and basic and primitive transformations. It uses a separate set of object-based analytics models with UML.

The fundamental differences between basic and applied models are closely related to analysis and design. Maintaining consistency of requirements models allows needs to be found through design and implementation. This is especially important when repeating the evolution as in the current exercise. it is easy to see the difference when the analysis and design models are distinctly different. when similar models are used in both activities, System developers need to be aware particularly and careful of the differences.

IV. Difference between Object-Oriented Analysis and Structured Analysis

The UML model for object-based analysis is different from case-specific analysis except for concept model for the application domain and the event domain. Here we can discuss the most important values about these models - system operating contracts, system sequence diagrams, use case descriptions and lastly use case diagrams.

There is a system-wide entity for defining usage requirements in UML. to complete a process, its use the sequence of state actions that occur when an actor - a system, an organization or an individual uses a system. In general, each use case is associated with an event. Its response to structural analysis is an important process associated with this event. Use case names are similar to the process name in parse used templates - usually a verb after an object.

Into the context schema UML lacks any support. unless the system is large, it contains a usage case diagram, showing all cases of use and equivalent to external units of structural analysis the actors included in each use case. Thus, a use case diagram does not show inputs or outputs it is like the rough equivalent of Diagram.

From the system inputs and outputs, the state list used replaces the listing analysis process as well as the data dictionary definitions. When they are well written, it can be easier for users to understand than structured forms. interaction between the subject or the actor and the system is detailed in primary use case. For each message from an actor, they have to show the expected internal response of the system and also should capture the detailed composition of the essential system inputs and outputs and its sequence.

With better level of detail in object-oriented analysis, a set of context diagrams would be replaced by system sequence diagrams. Based on a detailed primary usage case listing the system sequence diagrams are generated. As mention in the context diagram what is happening in the system is not mention. In theory, for each message there is a separate system sequence image, i.e., into the system there is separate system entry from an actor. a system sequence image may suffice for each usage case for the number of messages is small. As its name suggests, this graph will display the order in which the messages will come from the actor. The message format is similar to the action or function format, showing the list of its parameters and the name of the application's massage. These parameters are the basic input data elements.

An object head performs an operation with the message name when it receives a system message. The operating nodes of the system define the reaction of the process to the message of the actor, as shown

in the system sequence diagram. As an analysis model, it shows what needs to be done to respond to the response, not how to implement the response. This is achieved by writing a contract based on the terms of the publication. These posting terms are expressed in the form of a domain model; Shows cases of concepts and assemblies that have been added (or removed) to the domain model and have been modified the attribute values.

With the message in the system sequence diagram Each operating contract of this system is corelated and it turn is taken from the use case account. In order to get the required system response, the preconditions of the contract state must be right for the operation to be successful. As system operation's result of the implementation, domain model indicates the post conditions that required changes in the position.

V. CONCLUSION

Into this paper it is try to summaries the way to define the usefulness of object-oriented analysis and structured analysis. Still yet we are not decided every time that which method is best suitable for particular, and that's why we have to analyze application in both way and decide that which one is better suitable for the given application.

VI. REFERENCES

[1]. Rumbaugh J.,Blaha M.,Premerlani W.,Eddy F.,Lorensen W.: Object-Oriented Modeling and Design, Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.

[2]. Yourdon, E.: Modern Structured Analysis, Prentice-Hall Inc., Englewood Cliffs, NJ, 1989.

[3]. Larman, Craig (2005) Applying UML and Patterns: An Introduction to Object- Oriented Analysis and Design and the Unified Process, 3rd ed. Prentice-Hall, Upper Saddle River, NJ.

[4]. Yourdon, E. and ArgilIa, C. (1996). Case Studies in Object-oriented Analysis and Design. Londres, Prentice-Hall.

[5]. Yourdon, E. and ArgilIa, C. (1996). Case Studies in Object-oriented Analysis and Design. Londres, Prentice-Hall.

[6]. Boehm, B. W., & Bose, P. (1994). "A Collaborative Spiral Software Process Model Based on TheoryW." Proceedings, 3rd International Conference on the Software Process, Applying the Software Process (1994).

[7]. Dennis, Alan, Barbara Haley Wixom, and David Tegarden (2005) Systems Analysis and Design with UML 2.0: An Object- Oriented Approach. John Wiley & Sons, New York.

[8]. Jacobson , I., Booch, G. & Rumbaugh, J. (1999) The Unified Software Development Process. Addison-Wesley.

[9]. Rational Software White Paper TP026B, Rev 11/01. Rational Unified Process: Best Practices for Software Development Teams. Rational Software Corporation, 1998.

[10]. Baskerville, R., Fitzgerald, B., Fitzgerald, G., Russo, N. 1996, Beyond system development methodologies: time to leave the lamppost, in Orlikowski, W.J., Walsham, G., Jones, M.R., De Gross, J.I. (Eds),IT and Changes in Organisational Work, Chapman & Hall, London.

[11]. Höst, M., Wohlin, C., Thelin, T., 2005. Experimental context classification: incentives and experience of subjects, 27th International Conference on Software Engineering, St. Louis, Missouri, USA.

[12]. Jackson, M.A.: System Development, Prentice-Hall Inc., Englewood Cliffs, NJ, 1982.

[13]. Page-Jones, Meilir (1988) The Practical Guide to Structured Systems Design, 2nd ed. Yourdon Press, Englewood Hills, NJ.

[14]. Agarwal, R., De, P., and Sinha, A. P. 1999. Comprehending Object and Process Models:

An Empirical Study. IEEE Trans. Softw. Eng. 25, 4, 541- 556.

[15]. Goral Godhani, Maulik Dhamecha, “A Study on Movie Recommendation System Using Parallel MapReduce Technology”, IJEDR (2017)

[16]. Maulik Dhamecha, Dr. Amit Ganatra, Dr. C. K. Bhensadadiya, “Comprehensive Study of Hierarchical Clustering Algorithm and Comparison with Different Clustering Algorithms”, CiiT (2011)

[17]. Dhaval Chandarana, Maulik Dhamecha, “A Survey for Different Approaches of Outlier Detection in Data Mining”, IEEE (2015)

[18]. Maulik Dhamecha, Dr. Tejas Patalia, “Fundamental Survey of Map Reduce in Bigdata with Hadoop Environment”, Spinger – CCIS (2018)

[19]. Krishna Parmar, Nivid Limbasiya, Maulik Dhamecha, “Feature based Composite Approach for Sarcasm Detection using MapReduce”, IEEE (2018)

[20]. Maulik Dhamecha, Dr. Tejas Patalia, “MapReduce Foundation of Big data with Hadoop environment”, ELSEVIER - SSRN (2018)

[21]. Maulik Dhamecha, Krupa Dobaria, Dr. Tejas Patalia, “A Survey on Recommendation System for Bigdata using MapReduce Technology”, IEEE (2019)

[22]. Maulik Dhamecha, Dr. Tejas Patalia, “Comparative study of Dynamic Load Balancing algorithm in large scale data (Big data)”, IJAST (2020)

[23]. Object Management Group (OMG). UML Summary. (September 2001). http://www.dsi.uniroma1.it/~parisi/01-09-72.pdf .

Cite this article as : Maulik V. Dhamecha, "An Efficient Comparison between Structured Analysis and Object-Oriented Analysis", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 1, pp. 290-295, January-February 2021. Available at doi : https://doi.org/10.32628/CSEIT217369

Journal URL : https://ijsrcseit.com/CSEIT217369