Global Journal of Operating System and Techniques

Volume No. 9

Issue No. 1

January - April 2024

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

Global Journal of Operating System and Techniques

Aims and Scope

Global Journal of Computer Graphics & Techniques provides a medium to communicate information concerning interactive CG and CG applications. The journal focuses on interactive computer graphics, visualization and novel input modalities including virtual environments, and, within this scope, on graphical models, data structures, languages, picture manipulation algorithms and related software.

Global Journal of Operating System and Techniques

Global Journal of Operating System and Techniques

(Volume No. 9, Issue No. 1, Jan - Apr 2024)

Contents

Strengthening Collaborative Mind Map Tool To Facilitate Requirement Elicitation Process

Shahnawaz Ahmad and Yahiya Gazi
M. Tech. Scholar, Department of Computer Science and Engineering,
Faculty of Engineering and Technology, Al-Falah University,
Dhauj, Faridabad, Haryana, India.
E-mail: shahnawaz98976@gmail.com, yahiyaghazi@gmail.com

ABSTRACT

Requirements elicitation is a crucial task in any software development process. It is notable as a major contributor to the project failure. To minimize and subsequently improve the process, folding Mind Map concept in the elicitation process and implementing it collaboratively is being proposed in this study. The aim of the study is to study the impact of the Mind Map and groupware in facilitating requirements elicitation. A prototype based on Power Meeting is developed to illustrate the idea of mind map groupware called Collaborative Mind Map Tool (CMMT) which to be analyzed its impact towards requirement elicitation process.

Keywords: ***Requirements elicitation, folding Mind Map, Collaborative Mind Map Tool (CMMT)***

1. INTRODUCTION

Requirements elicitation is a process of identifying the requirements of the problem to be resolved by gathering, uncovering and understanding the needs of stakeholders. It involves the process of extracting and understanding the needs of stakeholders, and defining the application domain and problem context with explicit and precise descriptions. Requirements are obtained from the stakeholders as the primary resources, careful analysis of the organization, the application domain and business process where the system will be deployed (Kotonya, et al., 1998). Obviously those activities require intensive communication, collaboration and cooperation between stakeholders and requirement engineers. During this process, it is essential to capture the "right" understanding of the problem and interpret it correctly in agreeing representation. It has well been recognized that requirements elicitation is the most crucial and important aspect in any software development prior to analysis and specification stage.

Despite its crucial process, it is notable that requirement gathering is a major contributor to the failure in any software development project (Macaulay, 1996; Bohem, 1981; Rahman, 2004). Studies indicate that among the main problems in the requirements elicitation process are: gaps in communication and understanding, inadequacy of management; lack of knowledge and skilled people in approaching the requirement process; and process with wrong techniques and methods (Al-Rawas, et al.,1996; Weigers, 2003; Tsumaki, et al., 2005).

To overcome the situations, a Mind Map, which is a thinking "tool" that reflects how the information is stored and retrieved in a more organized and systematic way inside a human brain, will be bring into the requirement elicitation. Mind Map is a technique which uses graphical illustration in expressing thoughts and idea based on the concept of Radiant Thinking- a natural function of human mind (Buzan, et al., 1995). The subject of attention is placed in a central image, and radiates the ideas from the central image and hierarchically expands and associates the branches and its sub-branches with keywords. Information is categorized and classified, and subject is layout in a "snapshot" and organized way.

Furthermore, incorporating objects which are more stimulating than words such as shapes, images and colors in Research done by Ralph Haber, and later by R.S. Nickerson, have proven that images are more stimulating than words and thus generate more creative ideas and encourage better memorization. The Mind Map is used widely by many individuals and organizations and can be applied in most activities or situations, i.e. in decision making, analysis, problem solving, to-do list, note taking, brainstorming, presentation or even to write down simple notes.

Exploitation of Mind Map will generally multiply individual personal creative capabilities when collaborate in a group (Buzan, et al., 1995). Psychologically, group mind map encourages information sharing and teamwork (Morris, et al., 1998). It generates the sense of contribution which then encourages the team members to commit on job execution by viewing and understanding ultimate objective of the subject matter. This is essential criterion in teamwork to have same vision and mission (Hughes, 2008).

Greater exploitation can be achieved when incorporates the group mind map into groupware. Groupware is a technology that supports systematic and structured group collaboration (RAMA, et al., 2006). Generally, groupware is categorized into two primary dimensions of perspectives; the time and the space. The time illustrates the collaboration happens at same time (synchronous) or different time (asynchronous), whereas the space view on the perspective of where the collaboration is executed; whether it is in the same place (co-located) or different place (none co-located).

There are many significant advantages of groupware compared to a single user system, i.e. 1) To facilitate communication; make it faster, clearer and more persuasive 2) As a mechanism to transfer and share knowledge 3) To help in motivating individual to perform better 4) To collaborate a group of people of the same interest 5) A way to form structured group coordination and proactive collaboration 6) To save time and cost.

However, building groupware system is considered to be more challenging compared to individual system due to distinguish features in groupware system such as 1) presentation of groupware- high level of usability criteria; easy to learn, easy to use, error tolerant and subjectively pleasant, 2) collaboration and communication mechanism-classify the private and sharing information and maintain the consistency of the data throughout the instance, 3) work coordination- a defined process to be followed to encourage work accomplishment 4) openness- easily be integrated with other applications 5) group composition- management and control of the accessibility of groupware participants (Wells, 1996; Volksen, 1992)

Collabarative Mind Map Tool (Cmmt) in Facilatating Requirement Elicitation

Many studies have been done to find ways in improving requirement elicitation mainly due to the cognizance of the requirement engineering as one of the crucial factors contributing to the success of a software project. The process is more about people, communication and collaboration. A few studies and attempts to use Mind Map tool in various phase of software development process has been encountered (Eric T. Blue, 2006). However, very little usage of Mind Map in requirement activities as compared to project management activities. One convincing example that practically uses Mind Map in requirement elicitation is by Kenji Hiranabe. In his article, he has outlined few benefits and illustrated the example with specific template in using Mind Map in requirement process. He also provides a way on how to map mind map into UML diagrams using JUDE-a design and communication tool (Hiranabe, 2007; Change Vision, Inc, 2006).

Surveyed by Chucked Frey (2007) shows that collaboration is the most beneficial feature in Mind Map tool, and according to Nikos Drakos, a Gartner Inc. analyst, Mind Map will become more interesting and beneficial if it is move towards collaborative online (Gilhooly 2006). this indicates that collaborative Mind Map tool has potential significant impact to people and the process of software development in particular.

Requirement and Specification

Conceptually, the proposed CMMT hybrid the features from Mind Map theories and the distinguish features of groupware system; with the objective to improve the current problem facing in requirement elicitation.

The following are features proposed to be in CMMT:

Presentation of the tool: Inherited from the concept of Mind Map, the tool should tag with high level of usability. Conceptually, Mind Map is a simple and easy concept as it is based on how our brain works. Thus, the tool should be easy to integrate into practice and simple to use. Additionally, color, shapes and images will be used to trigger information which make the tool more attractive and motivate users to use it.

Communication mechanism: Radiant concept of mind map should articulate the information in a better way and present them in a more structure, clearer and more focus. With categorization and prioritization of information help to better understand the requirement and reduce ambiguity and inconsistency of requirements. Enriching the value of the tool, it should support both "synchronous" and "asynchronous" features. Having these features, existing requirement elicitation techniques (e.g. introspection, task analysis, brainstorming, laddering, requirements workshops, goal-based approach, scenarios, viewpoints and etc) can easily blend with mind map.

Collaboration mechanism: Information can be classified as private or sharing depending on the scenarios. A control mechanism is provided to manage the accessibility of information among the participants with centralize architecture to maintain data consistency. Nowadays, increasing in globalization has demanded multisite software development organizations (Damian, et al., 2003). Making the tool as a web based, it able to support distributed geographical requirement gathering activities which obviously save time and cost.

Work coordination: A floor control or session control is used to control the session and to ensure the process is executed in defined and structured manner. It also controls the accessibility of the participants in each elicitation session. This is to make the elicitation process more systematic, focus and efficient.

Requirement elicitation specific features: 1) Data Dictionary – A feature where crucial term or glossary can be defined. One of the problems in requirement elicitation is not sharing common understanding of concepts and terms. This ensures all crucial terms and concepts can be defined and agreed between stakeholders. 2) Record and Playback - A feature to record and replay each of the session. This can be used to further understand and review on the agreed requirements or decision. 3) Attachment Uploading – A feature where each requirement/information can be attached with supporting resources (e.g. organization plan, business process, legacy system manuals and etc.). This is to cater for which requirements may originate from different sources and formats. 4) Prioritization and Status updates – A feature where each requirement can be attached with priority number and status. This is to prioritize on

the crucial requirements and to set the status of requirements (e.g. to revisit and decide on the specific requirement.). 5) Chat – A feature where users can discuss on any issues rose during the requirement elicitation session.

Implementation Strategy

The prototype of CMMT is built on top of Power Meeting framework. Power Meeting, a work-piece (on going research) by Dr. Weigang Wang (2008), is a web based synchronous groupware framework which offers flexible, customizable and extensible groupware development environment. It basically provides with basic collaborative functionalities such as floor session control, user management, transaction management, text messaging, voice chat, session control, and also other existing groupware plug-in tools such as calendar tool, pincard board and presentation slides.

From a high level technical point of view, the Power Meeting framework is modelled based on Model (Shared Model) – View Controller (MVC) architecture and transactional replicate architecture employed from Common Ground toolkit, to provide with the basic groupware services such as user management, session and group management, replication and transaction management and persistence management (Wang, 2008). The implementation is built using AJAX technologies offered by Google Web Toolkit (GWT). AJAX is used to gain the benefit of instantaneous and faster response time, asynchronous partial updates and other rich user experience characteristics.

The flexibility and collaborative features furnish by Power Meeting framework has made the prototype implementation and deployment of CMMT easier. Figure 1, illustrates the CMMT on the Power Meeting interface.

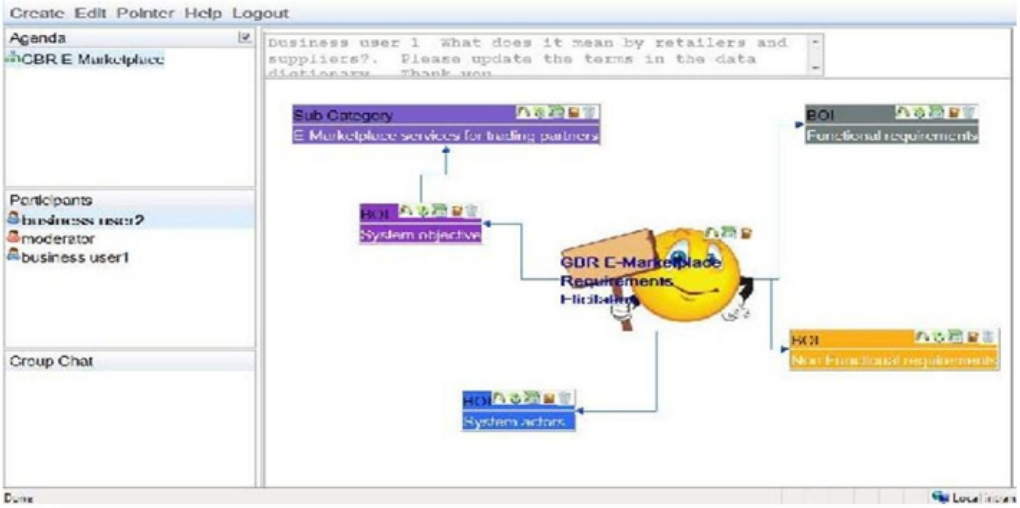


Figure 14: CMMT on Power Meeting

Mind Map Tool Architecture

Mind Map tool is plugged-in into the Power Meeting framework through the creation of Model/Shared Model objects and View-Controller objects. Model objects are used to hold and process the data and information in CMMT while the View-Controller objects hold the responsibilities to interpret the events initiated by the users, convey the instructions to the model for data processing or application specific processing and notify the view of any changes in the model object.

Below figure shows the object models of CMMT.

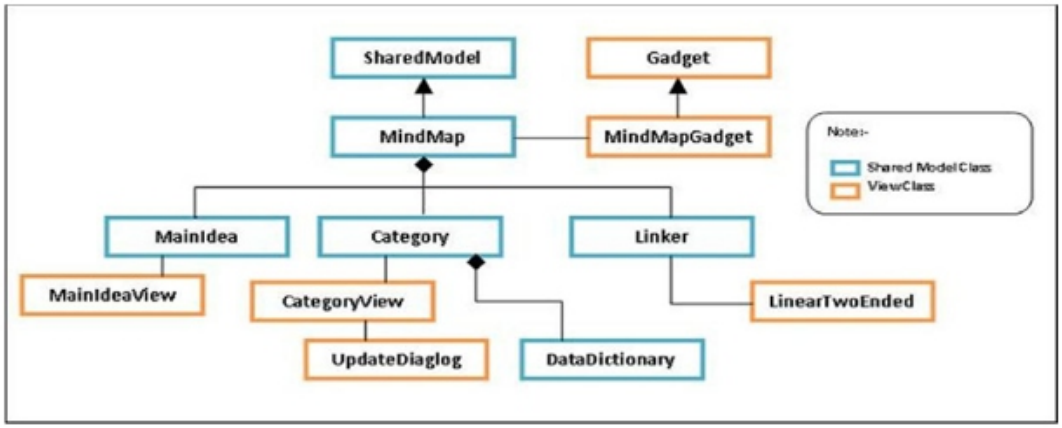


Figure 15: CMMT Object Model

Model objects are defined mainly on the basis of mind map theory by which it has the central subject matter (MainIdea) and radiates the idea into branches (Category) through associations (Linker). CMMT model objects need to extend SharedModel object provided by the framework to inherit the services to handle replication and transaction. Having this, CMMT model object instances can be made available and shared by all users in real time.

Evaluation and Analysis

The prototype-CMMT was successfully built which furnishes with three basic essential elements outlined to be achieved i.e. features that supporting requirement elicitation activities, Mind Map concept, and collaboration. A scenario of a software project development is given to a small group of participants to be executed for evaluation. The selected participants are those who are already familiar with elicitation process but having minimal background knowledge on mind map. Having this focus group we aim to get more reliable insight on the impact of CMMT towards elicitation process.

Feedback and comments were gathered through questionnaire at the end of the elicitation activities. The goal are to capture responses on usefulness of CMMT in requirement elicitation, the effortlessness in using CMMT, the layout and the performance of CMMT. Scale of 1(Strongly Agree) to 5(Strongly Not Agree) were used in capturing the responses. As an addition, participants can also plainly comments on a free text space provided.

More than 70% of participants agree that CMMT overlays a structured way of capturing requirements resulting to better understanding of the gathered requirements. However, only 30 % of respondents agree that CMMT promotes group work and encourages more detailed requirements capturing. In terms of the layout, over 60% responses in the group agree that the CMMT layout is easy to use and learn yet they are not sure on the effectiveness of the layout to generate creativity and memorization. On the other hands, the response time of CMMT is generally bearable to the participants. Since it was only tested by a small group of people, this finding would not be a strong claim to support the hypothesis. However, the work presented has provided an insight for future works in collaborative groupware and requirement engineering field.

Conclusion and Future Works

Observing the result from the evaluation and analysis, we believe that CMMT has demonstrated its potential capabilities in supporting requirement elicitation process.

In future, CMMT prototype is to be tested and validated in real software development project situation in order to get an actual insight of CMMT in terms of the significance offered and the improvement towards elicitation process. Also, further refinement should focus on implementing all the proposed features and subsequently extending the feature list so that it can fit to other requirement engineering processes; i.e. requirement modeling and analysis, requirement validation and requirement management.

References

[1]. Al-Rawas, A. & EasterBrook, S. (1996). Communication Problems in Requirement Engineering : A Field Study. London : Proceeding of the First Westminster Conference on Professional Awareness in Software Engineering. Royal Society.

[2]. Bohem, B. W. (1981). Software engineering economics. New Jersey : Prentice-Hall. Buzan, T, & Buzan, B. (1995). The Mindmap Book. London : BBC Books. Change Vision, Inc. (2006). JUDE/Professional Design & Modeling. Retrieved from http://jude.change-vision.com/jude-web/product/professional.html#mindmap_convert

[3]. Damian, Daniela E. & Zowghi, Didar. (2003). Requirement Engineering challenges in multi-site soft ware developmet organization. s.l. : Requirement Engineering Journal, 8.
[4]. Eric T. Blue. (2006). Mind Mapping and the Software Development Life Cycle. Eric's Blue Blog : Technology, Philosophy, and Personal Development. Retrieved from
[5]. http://eric-blue.com/2006/12/16/mind-mapping-and-the-software-development-life-cycle/
[6]. Gilhooly, K . (2006) Business on the map: Mind-mapping tools are finding a home in corporate IT.
[7]. Computerworld Inc. Retrieved from:http://www.computerworld.com/action/article.do?command=view ArticleBasic&taxonomyName =Software&articleId=112247&taxonomyId=18&pageNumber=1
[8]. Hiranabe, K. (2007) Agile Modelling with Mind Map and UML. Retrieved from StickyMinds.com website: http://www.stickyminds.com/sitewide.asp?ObjectId=11861&Function=DETAILBROWSE&ObjectType=ART
[9]. Hughes, Thomas B. (2008). What is Team Work and Why is it Important to Your Organization.
[10]. Retrieved from EzineArticles.com website: http://ezinearticles.com/?What-is-Team-Work-and-Why-is-it-Important-to-Your-Organization&id=1693070
[11]. Kotonya, G. & Sommerville, I. (1998). Requirements Engineering: Process and Techniques. New York : Willey.
[12]. Macaulay, Linda A. (1996). Requirement Engineering. London : Springer.
[13]. Morris, S, & Smith, J. (1998). Understanding Mind Maps in a week. London : Hodder & Stoughton.
[14]. Rahman, M. (2004). Requirement Engineering : UK e-Science Usability Task Force. [Internet] Oxford: Oxford e-Science Centre Review, Oxford University.
[15]. Rama, J. & Bishop, J. (2006). Survey and comparison of CSCW Groupware applications. Proceedingsof SAICSIT. s.l. : SAICSIT.
[16]. Tsumaki, T. & Tamai, T. (2005). A Framework for Matching Requirement Engineering Techniques to Project Characteristics and Situation Changes. Paris : Proceedings of SREP'05, Paris, France.
[17]. Völksen, G. (1992). Approach Strategies to Groupware. Retrieved from http://www.mathematik.unimarburg.de/~voelksen/publications/wmwork/www/approach_1.html
[18]. Wang, W. (2004). Power Meeting. Retrieved from http://sites.google.com/site/meetinginbrowsers/Home
[19]. Wang, W. (2008). Power Meeting on Common Ground: web based synchronous groupware with rich user experience. In Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective intelligence, Pittsburgh, PA, USA. WebScience '08. ACM, New York, NY, 35-39.
[20]. Retrieved from http://doi.acm.org/10.1145/1379157.1379166
[21]. Wells, D. (1996). Groupware & Collaboration Support. OBJS. Object Services and Consulting, Inc. Retrieved from http://www.objs.com/survey/groupware.htm
[22]. Wiegers, K.E. (2003). Software requirements : practical techniques for gathering and managing requirements throughout the product development cycle. [e-Book] s.l. : Redmond, Wash. :Microsoft Press, 2003. Available at: The John Rylands University Library Catalogue. Retrieved from http://www.netlibrary.com/Reader/

Study of Software Test box Generation Techniques

Mohit Saifi
M. Tech scholar, Computer Science and Engineering
Al-Falah School of Engineering and Technology. Dhouj, Faridabad, Haryana

A B S T R A C T

Software testing is an essential element of software development which sure the verification and validation method of the software. In arrange to do software testing we necessity have to apply the method of mapping the software for all its transition states and alone validating the output for a set of known input. For a any given part of software we will be writing a set of test cases that called test suites and it is used to collection together similar test cases. Test suites is a collection of test cases that are planned to be used to test a software program to demonstrate that it has some specific set of behaviors. In order to find out how a test case is valid or not for that we do not have specific device. We mostly depend on the software testers understanding of the obligation. The scope of this paper to study different technique use in test box, for example test box generation using genetic algorithm, test case generation using random based testing, test case generation using Model based testing. The test cases are derived by analyzing the dynamic behavior of the objects due to external and internal stimuli.

Keywords: ***Software Testing, Model based testing, Random Testing, Test box, Genetic algorithm.***

I. Introduction

Software testing is a significant movement in software development life cycle. Software organizations spend large segment of their budget in testing related behavior. Software Testing is a costly and time overriding process in software development life cycle. Automation of this phase may lead to overcome the above problems and also reduces the human effort in other ways it also helps in detecting the human intended errors and logical errors as well. The Automation of testing will not be that much productive in terms of cost and time consuming because if we have to wait till the end of the Software Development Life Cycle stage (SDLC). Software Testing[1] is any process or activity aimed at evaluating a system or attribute or capability of a program and determining through the purpose to find that whether it satisfies or meets the specified requirements or not. In simple words testing is executing a system in order to discover any errors, gaps or missing requirements in contrary to the actual requirements desire output.

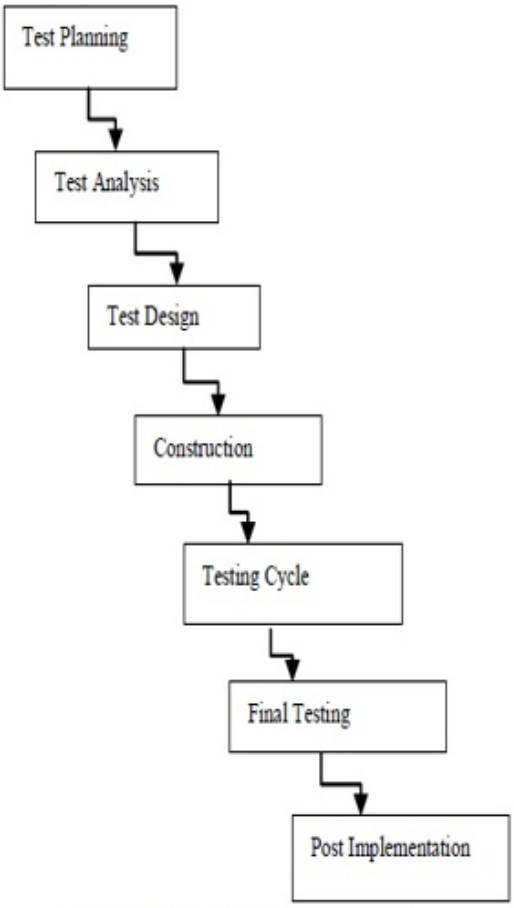


Figure1: Software Testing Steps

Test cases help the user to pen down entire coverage to the application and test all possible combinations in the application. It also provides the user to easily reproduce the steps that were undertaken to uncover a defect that as detected during test. It also provides the extent of which the testing has concluded and the areas in which the application is working fine. Throughout the years a huge number of different technique have been proposed for generating test cases. A test case is basically a description of a test. A Test case has the components that describe an input, event or action and expected response to determine if the feature of an application is working correctly. Test cases can be mapped directly to. Test cases are derived from use cases and can also be derived from system requirements. The one of the main advantages of generating test cases from requirements specifications and design will often help the software or test engineer to discover problems as early. As they can be created earlier in the development life cycle and get ready for use before the programs are constructed. Generating test cases early helps Software Engineers or test engineer can often find ambiguities and inconsistencies in the requirements specification and design documents. This will absolutely take down the cost of building the software systems as errors are eliminated early during the life cycle. The below diagram[2] highlights the steps of software testing.

II. Testing Stages

Test cases assist the user to pen down entire reporting to the application and test all possible combinations in the application. It also provides the user to easily reproduce the steps that were undertaken to uncover a defect that as detected during test. It also provides the extent of which the testing has concluded and the areas in which the application is working fine. Through the years a number of different methods have been proposed for generating test cases. A test case is a explanation of a test, independent of the way a given system is designed. Test cases can be mapped directly and derived from use cases. Test cases can also be derived from system requirements.. Additionally, when the test cases are generated early, Software Engineers can often find ambiguities and inconsistencies in the requirements specification and design documents. This will definitely get down the cost of building the software systems as errors are eliminated early during the life cycle. The below diagram[13] highlights the steps of software testing.

The above steps are under taken to test an application or software. Firstly the test plan is created which includes the overall structure of the plan i.e. requirements for testing, testing flow etc. Then the test cases are written according to some specific technique. There are many existing techniques are available. These cases are executed under suitable environment.

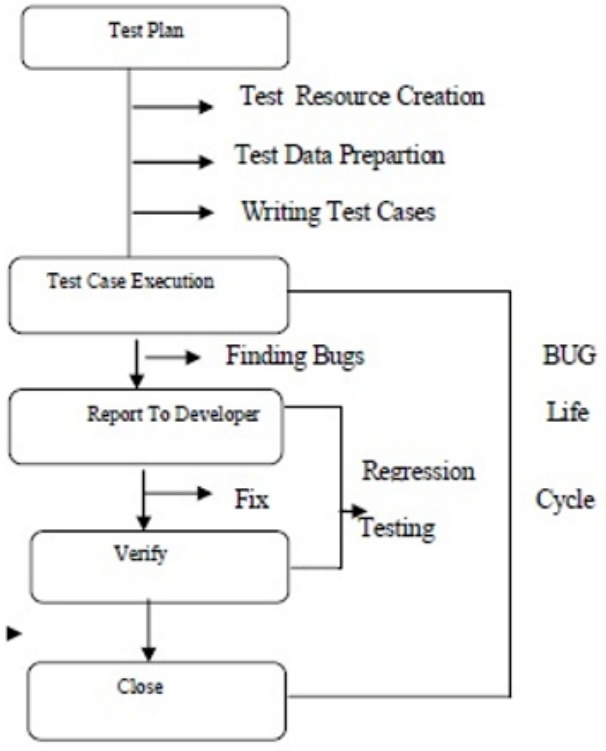


Figure 2 : Software Testing Stages

The bugs are found out and reported to the developers. The developer fixes these bugs and finally these are verified using some testing technique. This way the whole procedure works.

III. Software Testing Method

Method1:

Test Case generation using Genetic Algorithms (GA): In this technique, automated generation of test cases in object oriented systems has been presented. The test cases are derived by analyzing the dynamic behavior of the objects due to internal and external stimuli. The study is limited to the object diagrams taken from the UML (Unified Modeling Language) model of the system. In order to carry out all suitable test cases of a given object diagram, Genetic Algorithm's (GA) tree crossover has been proposed.

Method2:

Test Case generation using Random Testing : Random test case generation is a technique where in the test cases are generated not based on an algorithm but based on the ones assumption of the application. This technique is implemented on the above mentioned case study. The following classes will be tested and various test inputs will be provided to check for the faults. The framework used here to validate is called Auto Test and using this we will be able to predict the number of issues found and the number of detects undetected. The Auto Test frame work classifies test case s into the following categories: passed (no exception), unresolved (precondition violation in method under test), or failed (other exception). Method create test depicted below contains the main loop of the testing strategy used in this technique. At each step it selects a method for testing and then causes the execution of this method.

```
create_test (timeout):
from initialize pool
until timeout
loop
 m := choose (methods under test () )
create_test for method (m)
end
```

1. Method Initialize pool creates an empty pool of objects to be used for testing.
2. The Method methods under test returns the set of methods under test.
3. The non-deterministic method choose selects an arbitrary element of a set or a list.
4. The method create test for method is responsible for generating a call to method m.

Method 3:

Test box generation using a combination of Activity Diagrams and Sequence diagrams

Category Partition Method is simply a specification based testing technique with respect to some specific criteria. CPM first decomposes the functional specification into functional units and then examines each functional unit. It finds the categories for each parameter and environmental condition. It helps in identifying the parameters and environmental conditions that affect the execution behavior of the function. CPM is also helpful in finding the categories of information that characterize each parameter and environmental condition. We will first generate test scenarios from activity diagrams, which achieve path coverage criteria perfectly, followed by generation of test cases by analyzing the respective sequence and class diagrams of each scenario. This technique helps to reuse the design. Complex tests are built up by designing a test that runs through a series of use cases. First the test scenarios will be generated followed by analysis of each scenario using the sequence diagrams to formulate the test cases. Following gives a pictorial representation of the approach followed.

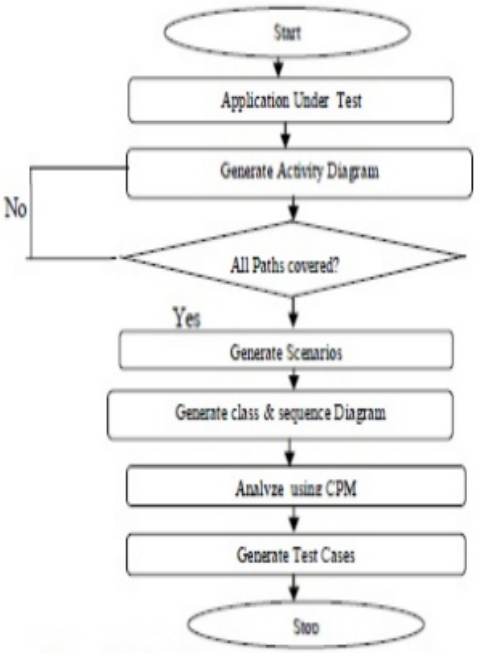


Figure 3. Pictorial Representation of Test scenario

Method 4:

Test Case generation using Model Based Testing Model Based Testing (MBT) is a black-box testing technique where common testing tasks such as test case generation and test result evaluation are automated based on a model of the application under test. This approach has recently spread to a variety of software domains but originates from hardware testing, most notably from telephone switches, and from the increasing use of object orientation and models in software design and software development. Model Based Testing (MBT) automate the complete design of test cases and the generation of the traceability matrix, which traces the link between requirements and generated test cases. Instead of

writing hundreds of test cases, the test designer constructs an abstract model of the system under test. The MBT tool is used to generate a set of test cases from that model. Below describes the manner in which the MBT takes place.

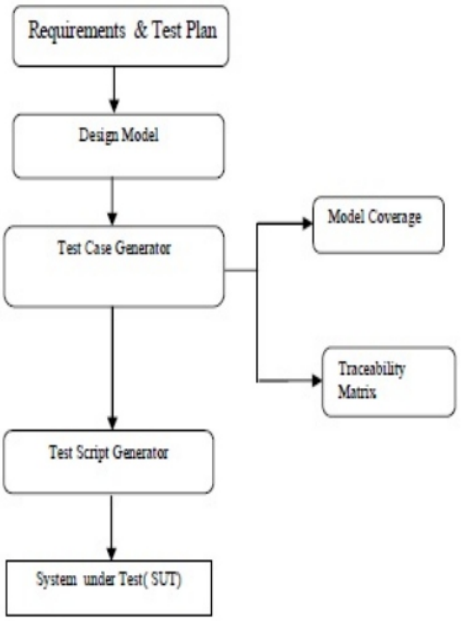


Figure4. Model based Testing Process Diagram

IV. Conclusion

We create various test box technique in a straight line from UML behavioral diagram, where the design is reused. By using our move toward defects in the design model can be detected during the study of the model itself. So, the defects can be removed as early as possible, thus reducing the cost of defect removal. First we generate test case generation using genetic algorithm, random based testing, model based testing and test scenarios from the activity diagram and then for each scenario the corresponding sequence diagram generated .After analyzing each category, its significant values and constraints are generated and respective test cases are derived. Test coverage criteria achieved is another advantage of our approach.

References

[1]. Aljahdali S. H , Ghiduk A. S and Telbemy M.(2010), The limitations of Genetic algorithms in software testing, International Conference on Computer Systems and Applications, held at Hammamet from 16th May to 19th May 2010, ISBN: 978-1-4244-7716-6, pp: 1-7..

[2]. Jin-Cherng Lin and Pu-Lin Yeh, (2001), Automatic Test Data Generation for Path Testing Using Genetic Algorithms, International Journal of Information Sciences, January 2001, New York, USA, Volume 131, Issue 1-4,.

[3]. Rudnick Elizabeth M., Greenstein Gary S., "A Genetic Algorithm Framework for Test Generation" IEEE Transactions on Computer Aided design of Integrated Circuits and Systems, VOL. 16, NO. 9, 1997.

[4]. R.P. Pargas, M.J. Harrold, and R.R. Peck, "Test data generation using genetic algorithms", The Journal of Software Testing, Verification and Reliability, Vol.9, No.4, Pp. 263 – 282, 199.

[5]. T.Y. Chen, Pak-Lok Poon and T.H Tse," A choice Relation Framework for supporting category-partition Test case Generation", IEEE Transactions on software engineering Vol. 29, No.7, 2003.

[6]. Ghiduk, Ahmed S and Girgis, Moheb R.(2010), Using Genetic Algorithms and Dominance Concepts for Generating Reduced Test Data, Informatica (Slovenia), Volume 34, Number 3, pp.377-385.

[7]. Sarma Monalisa and Mall Rajib, "Automatic Test Case Generation from UML Models," 10th International Conference on Information Technology, pp. 196-201, 2007.

[8]. Ilinca Ciupa, Alexander Pretschner, Andreas Leitner, Manuel Oriol, Bertrand Meye: "On the Predictability of Random Tests for Object-Oriented Software" International Conference on Software Testing, Verification, and Validation, 2008.

[9]. Reza, H., K. Ogaard and A. Malge," A model based testing technique to test web applications using state charts", Proceedings of 5th International Conference on Information Technology: New Generations, pp. 183-188, 2008.

[10]. Prasanna M., Chandran K.R., "Automatic Test Case Generation for UML Object diagrams using Genetic Algorithm",Vol.1,No.1,2009.

[11]. Singh Kulvinder and Kumar Rakesh, "Optimization of Functional Testing using Genetic Algorithms" International Journal of Innovation, Management and Technology, Vol. 1,No. 1, 2010.

[12]. Sthamer "The automatic generation of software test data using genetic algorithms". Phd thesis, university of Glamorgan, Pontyprid, wales, Great Britain, 1996.

Effective Regression Testing Technique is use to Reducing the Residual Defects

Mohd. Muneer Siddiqui
M. Tech scholar, Computer Science and Engineering
Al-Falah School of Engineering and Technology. Dhouj, Faridabad, Haryana

A B S T R A C T

The principle of regression testing is to make sure that changes made to software, such as adding new features or modifying existing features, have not adversely affected features of the software that should not change. Many techniques have been reported on how to select regression tests so that the number of test cases does not grow too large as the software evolves. Our technique combines modification, minimization and prioritization-based selection using a list of source code changes and the execution traces from test cases run on previous versions. One important issue associated with a system lifetime view that we have overlooked in past years is the effects of residual defects (persist undetected) across several releases of a system. Depending on an organization's business goals and the type of system being built, residual defects might affect the level of success of the software products. In this paper, we conducted an empirical study to investigate whether Regression Testing Technique are effective in reducing the persistence of residual defects across a system's lifetime considering test case prioritization techniques.

Regression testing is a testing activity that is performed to provide that changes do not harm the existing behavior of the software. A number of deferent approaches have been studied to maximize the value of the accrued test suite: minimization, selection and prioritization. Test suite minimization seeks to eliminate redundant test cases in order to reduce the number of tests to run. Test case selection seeks to identify the test cases that are relevant to some set of recent changes. Test case prioritization seeks to order test cases in such a way that early fault detection is maximized. This paper surveys each area of minimization, selection and prioritization technique and discusses open problems and potential directions for future research.

Keywords: ***Regression Testing, Modification-Based Test Selection, Test Set Minimization, test case prioritization, residual defects.***

1. Introduction

Regression Testing is a well-recognized practice in the software industry. Many techniques have been proposed to perform cost-effective regression testing. However, one type of defect was not being studied experts in this field. That defect type is residual defects - those that have existed through several releases undetected. This paper looks to study the effectiveness of using regression testing to find this type of defect.

The paper states that residual defects could make about 22% of the total defects within a piece of software. The cost of fixing residual defects grows with program complexity.

It is a complex procedure that is all the more challenging because of some of the recent trends in software development paradigms.

For example, the component-based software development method tends to result in use of many black-box components, often adopted from a third-party. Any change in the third-party components may interfere with the rest of the software system, yet it is hard to perform regression testing because the internals of the third-party components are not known to their users.

The shorter life-cycle of software development, such as the one suggested by the agile programming discipline, also imposes restrictions and constraints on how regression testing can be performed within limited resources.

Naturally, the most straightforward approach to this problem is to simply execute all the existing test cases in the test suite; this is called a retest-all approach. However, as software evolves, the test suite tends to grow, which means it may be prohibitively expensive to execute the entire test suite.

Testers might rerun all test cases generated at earlier stages to ensure that the program behaves as expected. However, as a program evolves the regression test set grows larger, old tests are rarely discarded, and the expense of regression testing grows. Repeating all previous test cases in regression testing after each minor software revision or patch is often impossible due to the pressure of time and budget constraints. On the other hand, for software revalidation, arbitrarily omitting test cases used in r egression testing is risky. In this paper, we investigate methods to select small subsets of effective fault-revealing regression test cases to revalidate software.

In this paper, we first select tests from the regression testing that execute any of the modifications in the old program. Third section includes case studies.

2. Regression Testing

Regression testing is performed between two deferent versions of software in order to provide confidence that the newly introduced features of the System under Test (SUT) do not interfere with the existing features. While the exact details of the modifications made to SUT will often be available, they may not be easily available in some cases. For example, when the new version is written in a different programming language or when the source code is unavailable, modification data will be unavailable.

The following notations are used to describe concepts in the context of regression testing. Let P be the current version of the program under test, and P0 be the next version of P. Let S be the current set of specifications for P, and S0 be the set of specifications for P0. T is the existing test suite. Individual test cases will be denoted by lower case: t. P (t) stands for the execution of P using t as input.

2.1. Test case prioritization

Test case prioritization approaches that are used to execute the regression testing in a cost-effective manner were investigated. We discussed the critical issues and best practices that a software company should focus on before and after the implementation of test case prioritization techniques inside the company. Due to the increasing complexity of today's software intensive systems, the number of test cases in a software development project increases for an effective validation & verification process and the time allocated to execute the regression tests decreases because of the marketing pressures. For this reason, it is very crucial to plan and setup test case prioritization infrastructures properly in software companies to improve the software testing process. Ten best practices for a successful test case prioritization are introduced and explained in this study.

2.2. Regression Test Suite

A regression test suite of 1000 distinct tests was created based on the operational profile of how the space program was used. An operational profile, as formalized by Musa and used in our experiment, is a set of the occurrence probabilities of various software functions. To obtain an operational profile for space we identified the possible functions of the program and generated a graph capturing the connectivity of these functions. Each node in the graph represented a function. Two nodes, A and B, were connected if control could flow from function A to function

2.3. Economic model for Regression Testing

Software engineering methodologies are subject to complex cost-benefit tradeoffs. Economic models can help practitioners and researchers assess methodologies relative to these tradeoffs. Effective economic models, however, can be established only through an iterative process of refinement involving analytical and empirical methods. Sensitivity analysis provides one such method. By identifying the factors that are most important to models, sensitivity analysis can help simplify those models; it can also identify factors that must be measured with care, leading to guidelines for better test strategy definition and application. In prior work we presented the first comprehensive economic model for the regression testing process, that captures both cost and benefit factors relevant to that process while supporting evaluation of these processes across entire system lifetimes. In this work we use sensitivity analysis to examine our model analytically and assess the factors that are most important to the model. Based on the results of that analysis, we propose two new models of increasing simplicity. We assess these

models empirically on data obtained by using regression testing techniques on several non-trivial software systems. Our results show that one of the simplified models assesses the relationships between techniques in the same way as the full model.

3. A Case Study

A case study was conducted on a space program developed for the European Space Agency.4 A modification based selection technique and the subsequent test set minimization and prioritization techniques were used to select regression tests for re execution. Three metrics, size reduction, recall and precision, were computed to measure the cost-effectiveness of these techniques. Our experiment is explained below in detail.

3.1. The Space Program

The space program provides a language-oriented user interface that allows the user to describe the configuration of an array of antennas using a high level language. Its purpose is to prepare a data file in accordance with a predefined format and characteristics from a user, given the array antenna configuration described in a language-like form.

An appropriate Array Definition Language was defined and used within the program. This language allows the user to describe a certain antenna array by a few statements instead of having to write the complete list of elements, positions and excitations. The program consists of about 10,000 lines of code divided into three subsystems, parser, computation, and formatting. Details of these subsystems can be found in their design documents.

The fault set for the space program used in this study was obtained from the error-log maintained during its testing and integration phase (see Appendix). For convenience, each fault has been numbered as Fk where the integer k denotes the fault number. This number does not indicate the order in which faults were detected.

3.2. Regression Test Suite

A regression test suite of 1000 distinct tests was created based on the operational profile of how the space program was used. An operational profile, as formalized by Musa and used in our experiment, is a set of the occurrence probabilities of various software functions. To obtain an operational profile for space we identified the possible functions of the program and generated a graph capturing the connectivity of these functions. Each node in the graph represented a function. Two nodes, A and B, were connected if control could flow from function A to function B.

There was a unique start and end node representing functions at which execution began and terminated, respectively.

A path through the graph from the start node to the end node represents one possible program execution. To estimate the occurrence probability of the software functions, each arc was assigned a transition probability, i.e., the probability of control flowing between the nodes connected by the arc.

4. Test Case Prioritisation

Test case prioritization seeks to find the ideal ordering of test cases for testing, so that the tester obtains maximum benefit, even if the testing is prematurely halted at some arbitrary point. The approach was first mentioned by Wong et al.

However, in that work it was only applied to test cases that were already selected by a test case selection technique. Harrold and Rothermel proposed and evaluated the approach in a more general context.

4.1. Coverage-based Prioritization

Structural coverage is a metric that is often used as the prioritization criterion. The intuition behind the idea is that early maximization of structural coverage will also increase the chance of early maximization of fault detection. Therefore, while the goal of test case prioritization remains that of achieving a higher fault detection rate, prioritization techniques actually aim to maximize early coverage.

Rothermel et al. reported empirical studies of several prioritization techniques [144,145]. They applied the same algorithm with different fault detection rate surrogates. The considered surrogates were: branch-total, branch-additional, statement total, statement-additional, Fault Exposing Potential (FEP)-total, and FEP-additional.

The branch-total approach prioritizes test cases according to the number of branches covered by individual test cases, while branch-additional prioritizes test cases according to the additional number of branches covered by individual test cases.

The statement-total and statement-additional approaches apply the same idea to program statements, rather than branches.

Algorithmically, `total' approaches are essentially instances of greedy algorithms whereas `additional' approaches are essentially instances of additional greedy algorithms.

The FEP of a test case is measured using program mutation. Program mutation introduces a simple syntactic modification to the program source, producing a mutant version of the program [23]. This mutant is said to be killed by a test case if the test case reveals the deference between the original program and the mutant. Given a set of mutants, the mutation score of a test case is the ratio of mutants that are killed by the test case to the total kill-able mutants. The FEP-total approach prioritizes test cases according to the mutation score of individual test cases, while the FEP-additional approach prioritizes test cases according to the additional increase in mutation score provided by individual test cases. Note that FEP criterion can be constructed to be at least as strong as structural coverage; to kill a mutant, a test case not only needs to achieve the coverage of the location of mutation but also to execute the mutated part with a set of test inputs that can kill the mutant.

Elbaum et al. extended the empirical study of Rothermel et al. by including more programs and prioritisation surrogates.

Among the newly introduced prioritization surrogates, function-coverage and function-level FEP enabled Elbaum et al. to study the effects of granularity on prioritization. Function-coverage of a test case is calculated by counting the number of functions that the test case executes. Function-level FEP is calculated, for each function f and each test case t, by summing the ratio of mutants in f killed by t. Elbaum et al. hypothesized that approaches with coarser granularity would produce lower APFD values, which was confirmed statistically.

Jones and Harrold applied the greedy-based prioritization approach to Modified Condition/Decision Coverage (MC/DC) criterion [82]. MC/DC is a `stricter form' of branch coverage; it requires execution coverage at condition level. A condition is a Boolean expression that cannot be factored into simpler Boolean expressions. By checking each condition in decision predicates, MC/DC examines whether each condition independently affects the outcome of the decision [28]. They presented an empirical study that contained only an execution time analysis of the prioritization technique and not an evaluation based on fault detection rate.

4.2. Interaction Testing

Interaction testing is required when the SUT involves multiple combinations of different components. A common example would be configuration testing, which is required to ensure that the SUT executes correctly on different combinations of environment, such as different operating systems or hardware options. Each component that can be changed is called a factor; the number of choices for each factor is called the level of the corresponding factor. As the number of factors and levels of each factor increase, exhaustive testing of all possible combinations of factors becomes infeasible as it requires an exponentially large test suite.

Instead of testing exhaustively, pair-wise interaction testing requires only that every individual pair of interactions between different factors are included at least once in the testing process. The reduction grows larger as more factors and levels are involved. More formally, the problem of obtaining interaction testing combinations can be expressed as the problem of obtaining a covering array, CA (N; t; k; v), which is an array with N rows and k columns; v is the number of levels associated with each factor, and t is the strength of the interaction coverage (2 in the case of pair-wise interaction testing).

Bryce and Memon also applied the principles of interaction coverage to the test case prioritization of Event-Driven Software (EDS). EDS takes sequences of events as input, changes state and outputs new event sequences. A common example would be GUI-based programs. Bryce and Memon interpreted t-way interaction coverage as sequences that contain different combinations of events over t unique GUI windows. Interaction coverage based prioritization of test suites was compared to deferent prioritization techniques such as unique event coverage (the aim is to cover as many unique events as possible, as early as possible), longest to shortest (execute the test case with the longest event sequence _rst) and shortest to longest (execute the test case with the shortest event sequence first). The empirical evaluation showed that interaction coverage based testing of EDS can be more efficient than the other techniques, provided that the original test suite contains higher interaction coverage. Note that Bryce and Memon did not try to generate additional test cases to improve interaction coverage; they only considered permutations of existing test cases.

5. Conclusion

Effective regression testing is a trade-off between the number of regression tests needed and the cost.
In this paper, we propose a modification-based technique followed by test set minimization or prioritization to determine which regression tests should be rerun.

Three metrics, size reduction, precision, and recall, are used to examine the goodness of using (I) and (II). These metric values depend not only on the nature of the regression test suite but also the extent and the locations of the modifications. Unlike many other proposed test selection techniques, ours is supported by a tool called ATAC.

If for reasons of safety or performance an application is not tolerant of a great deal of intrusive instrumentation, our modification-based test selection can be conducted at a higher level of granularity such as at the function or subsystem level.

On the other hand, if a high percentage of precision and recall is required, one can apply more complicated, and more expensive, techniques such as relevant slicing to construct a smaller super set, 0, of those regression tests which need to be re executed.

We can then apply test set minimization and prioritization to 0 to identify a representative subset of tests which have a higher priority to be rerun.

The bottom-line for making a decision on whether any of these alternatives should be adopted goes back to the original trade-off problem: What we should do in regression testing versus what we can afford to do. Experiments are underway to compare the cost-effectiveness of these alternatives.

The results of these studies will provide more information to help us determine how to conduct efficient and effective regression testing in practice.

6. References

[1] IEEE Standard Glossary of Software Engineering Terminology. IEEE Press, 10 Dec 1990.
[2] H. Agrawal, J. R. Horgan, E. W. Krauser, and S. A. London. Incremental regression testing. In Proceedings of the International Conference on Software Maintenance (ICSM 1993), pages 348{357. IEEE Computer Society, September 1993.
[3] N. Alshahwan and M. Harman. Automated session data repair for web application regression testing. In Proceedings of 2008 International Conference on Software Testing, Veri_cation, and Validation, pages 298{307, Los Alamitos, CA, USA, 2008. IEEE Computer Society.
[4] J. H. Andrews, L. C. Briand, and Y. Labiche. Is mutation an appropriate tool for testing experiments? In Proceedings of the 27th International Conference on Software Engineering (ICSE 2005), pages 402{411. ACM Press, May 2005.
[5] R. Anido, A. R. Cavalli, L. P. Lima Jr, and N. Yevtushenko. Test suite minimization for testing in context. Software Testing, Veri_cation and Reliability, 13(3):141{155, 2003.
[6] T. Apiwattanapong, R. Santelices, P. K. Chittimalli, A. Orso, and M. J. Harrold. Matrix: Maintenance-oriented testing requirements identi_er and examiner. In Proceedings of Testing: Academic & Industrial Conference on Practice And Research Techniques, pages 137{146, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
[7] T. Ball. On the limit of control ow analysis for regression test selection. In Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 1998), pages 134{142. ACM Press, March 1998.
[8] S. Bates and S. Horwitz. Incremental program testing using program dependence graphs. In Proceedings of the 20th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 384{396. ACM Press, January 1993.
[9] Y. Chen, D. Rosenblum, and K. Vo, "TestTube: A system for selective regression testing," in Int'l. Conf. Softw. Eng., May 1994, pp. 211-220.
[10] J. Offutt, J. Pan, and J. M. Voas, "Procedures for reducing the size of coverage-based test sets," in Proc. Int'l. Conf. Testing Comp. Softw., Jun. 1995, pp. 111-123.
[11] G. Rothermel and M. J. Harrold, "A safe, efficient regression test selection technique," ACM Trans. Softw. Eng. Meth., vol. 6, no. 2, pp. 173-210, Apr. 1997. (Pubitemid 127684922)
[12] G. Rothermel, R. Untch, C. Chu, and M. J. Harrold, "Test case prioritization," IEEE Trans. Softw. Eng., vol. 27, no. 10, pp. 929-948, Oct. 2001.
[13] A. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. Roos, "Time-aware test suite prioritization," in Int'l. Symp. Softw. Test. Anal, Jul. 2006, pp. 1-12.
[14] H. Do, S. Mirarab, L. Tahvildari, and G. Rothermel, "An empirical study of the effect of time constraints on the cost-benefits of regression testing," in Proceedings of the ACM SICSOFT Symposium on Foundations of Software Engineering, Nov. 2008, pp. 71-82.
[15] S. Elbaum, A. G. Malishevsky, and G. Rothermel, "Test case prioritization: A family of empirical studies," IEEE Trans. Softw. Eng., vol. 28, no. 2, pp. 159-182, Feb. 2002.
[16] J. Kim and A. Porter, "A history-based test prioritization technique for regression testing in resource constrained environments," in Int'l. Conf. Softw. Eng., May 2002, pp. 119-129.
[17] A. Orso, N. Shi, and M. J. Harrold, "Scaling regression testing to large software systems," in Proceedings of the International Symposium on Foundations of Software Engineering, Nov. 2004.
[18] N. Fenton and S. Pfleeger, "Science and substance: A challenge to software engineers," IEEE Software, pp. 86-95, Jul. 1994.

Revolutionizing Real Estate by Using Big Data Analytics

[1]Pankaj Roy [2]Dr. Ruchi Agarwal
Dept. of Computer Science School of Engineering and Technology
Sharda University, Greater Noida, India
er.pankajroy@gmail.com ruchi.agarwal@sharda.ac.in

A B S T R A C T

Big data is changing how businesses do business. Data is growing at an explosive rate. Data is generated through many sources like Social media, sensor data, spatial coordinates, RFID's and web-servers, etc. and remains in structured as well as unstructured form. Processing or analyzing the huge amount of data or extracting meaningful information is a challenging task. The term "Big Data" is used for high-volume, high-velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization [1]. Reasons for the data explosion are largely due to new technologies generating and collecting vast amounts of data. Much of the data generated is un-structured or semi-structured format, that makes it difficult to capture, storage, search, sharing, manage and process. To challenge the problems Hadoop an open source framework from Apache software foundation to store and process large data sets by using distributed model technique. This paper focuses on the application of "Big data analytics" in real estate market. With the use of meaningful extraction of information from big data can change the prospects of market available in real estate.

Keywords: Big data, big data analytics, real estate, hadoop, map reduce, HDFS, mongo db.

I. Introduction

Big data describes any voluminous amount of structured, semi-structured and un-structured data. Big data can be characterized by 3Vs: the extreme volume of data, the wide variety of types of data and the velocity at which the data must be processed [2]. Although big data doesn't refer to any specific quantity, the term is often used when speaking about petabytes and Exabyte of data, much of which cannot be integrated easily. This flood of data is generated by connected devices— from social networking sites and smart phones to sensors such as RFID readers and handheld smart devices, complex scientific sensors, retail, inventory and sales tracking devices etc. It comes in many formats, like text, document, image, video, and more. The real value of big data is not the storing of data but the value of big data is in the insights it produces when analyzed for getting some patterns, derived meaning, and decision making on the basis of history and producing some results which acts as a key ingredient for success in businesses. When implementing big data in organizations, the 3V's fall short, and a fourth V for value must become the driving focus. Unlocking the value in the data is the key to providing value to the business.

History:

In 1997 Michael Lesk publishes “How much information is there in the world?” Lesk concludes that “There may be a few thousand petabytes of information all told; and the production of tape and disk will reach that level by the year 2000. So in only a few years, (a) we will be able [to] save everything–no information will have to be thrown out, and (b) the typical piece of information will never be looked at by a human being.” James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers of the McKinsey Global Institute publish (in May 2011) “Big data: The next frontier for innovation, competition, and productivity.” They estimate that “by 2009, nearly all sectors in the US economy had at least an average of 200 terabytes of stored data (twice the size of US retailer Wal-Mart's data warehouse in 1999) per company with more than 1,000 employees” and that the securities and investment services sector leads in terms of stored data per firm. In total, the study estimates that 7.4 Exabyte's of new data were stored by enterprises and 6.8 Exabyte's by consumers in 2010. In May 2012 danah Boyd and Kate Crawford publish “Critical Questions for Big Data” in *Information, Communications, and Society*. They define big data as “a cultural, technological, and scholarly phenomenon that rests on the interplay of: (1) Technology: maximizing computation power and algorithmic accuracy to gather, analyze, link, and compare large data sets. (2) Analysis: drawing on large data sets to identify patterns in order to make economic, social, technical, and legal claims. (3) Mythology: the widespread belief that large data sets offer a higher form of intelligence and knowledge that can generate insights that were previously impossible, with the aura of truth, objectivity, and accuracy.” In 2010 Eric Schmidt speaks at the Techonomy conference in Lake Tahoe in California and he states that "there were 5 Exabyte's of information created by the entire world between the dawn of civilization and 2003. Now that same amount is created every two days.”

An Open Source Solution to the Big Data Explosion is Hadoop. Hadoop was created in 2006 out of the necessity for new systems to handle the explosion of data from the web. Free to download, use, enhance and improve, Hadoop is a 100% open source way of storing and processing data that “enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data, and can scale without limits” [3]. Big Data analysis is a technology that searches useful information such as a relation rule, a hidden value, and pattern from huge data [4].Commercial Real Estate firms face a basic challenge: how to provide clients with fast and insightful analyses that mitigate risk and maximize return on real estate investments. We are proposing a recommender system model which will help to fill the gap between user and firms. The framework is based on hadoop, which is a framework for running applications on large clusters built of commodity hardware. Hadoop comprises of two major components that is, HDFS (Hadoop Distributed File System) and Map Reduce.

The rest of the paper is organized as follows: in section II literature survey about the proposed work, in section III the brief description of the components of Hadoop, section IV, presents the Methodology & approaches used for big data analysis, section V presents the application of big data in real estate and analyze the problem and propose the proposed solution, and in last section VI, conclusion and future work of the paper.

II Literature

Companies are looking to a variety of data types and new forms of analysis in order to remain competitive. Forward looking companies are developing analytics ecosystems that make use of disparate kinds of data, including text data, social media data, machine data, and more. Figure 1 shows few of the big data challenges.

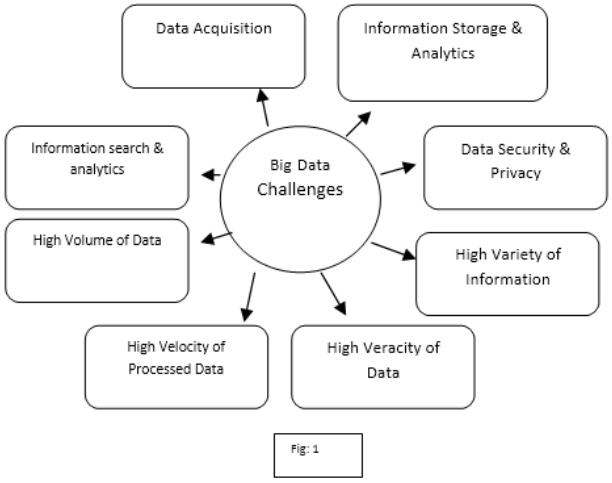


Fig: 1

McKinsey that first puts forward the concept of big data. Big data has become an important factor of production permeating into different industries and functional areas for now. The mining and applications of large data means a new wave of productivity growth and consumer surplus [5]. The Nature made big data special issue as early as 2008, describing the potential value of big data and the challenge from data handling techniques in Internet technologies, biomedical and environmental sciences, cloud technologies and other fields. Likewise, the Science, one of the top international academic journals, published a special issue to convey the ideas of the time of big data. In 2012, the European academy of informatics and mathematics studied big data systematically including the management of big data, the direction and results of academic research [6]. Brown B and Chui M etal. Believe that the reaching of big data provides potential for both realty enterprises and realty buyers with direct data sharing, bypassing the estate agents. And this can be a shock to the sense of data property right [7]. Every day, 2.5 quintillion bytes of data are created and 90% of the data in the world today were produced within the past two years [8] *(IBM 2012)*. Many new tools and technologies have arisen over

the past several years to process Big Data. Although the specific technologies and applications for Big Data vary widely, one of the major enablers underlying many Big Data platforms is the Map Reduce computational paradigm. Many Map Reduce applications come from e-commerce and social media companies such as Amazon [9], Google [10, 11], Facebook [12], and Twitter. Hadoop, one of the most widely used Big Data analytics platforms today, has Map-Reduce as its basis [13].

III Hadoop

Apache Hadoop is a framework that allows for the distributed processing of large datasets across clusters of commodity computers using a simple programming model. It is an Open-source Data Management with scale-out storage & distributed processing. In year 2004, Google published a white paper on Map Reduce. Map Reduce framework provides a parallel processing model and associated implementation to process huge amount of data and map reduce framework was adopted by an open source project called Hadoop. Scalable, reliable, economical, flexible, robust ecosystem and real time are few of Hadoop key of characteristics. Hadoop is designed to process large volumes of information by connecting many commodity computers together to work in parallel in efficient manner [14]. Core components of Hadoop (Fig: 2). HDFS- Hadoop distributed file system (storage) and Map-Reduce (processing).

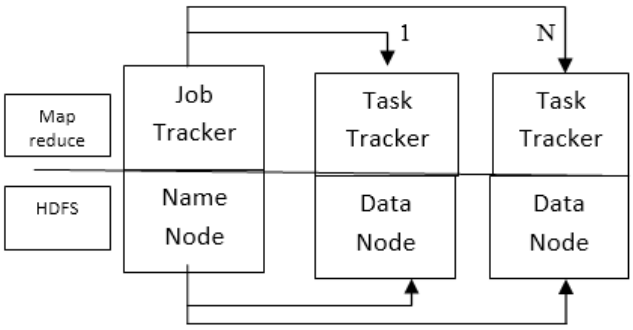


Fig: 2

HDFS: HDFS has Master/slave architecture. An HDFS cluster consists of a single Name node, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of Data nodes, usually one per node in the cluster, which manages storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of Data nodes. The Name node executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to Data nodes. The Data nodes are responsible for serving read and write requests from the file system's clients. The Data nodes also perform block creation, deletion, and replication upon instruction from the Name node. Some of the goals of HDFS are Very Large Distributed File System, Assumes Commodity Hardware and Optimized for Batch Processing, Runs on heterogeneous OS.

IV Methodology & Approaches

Hadoop's components:

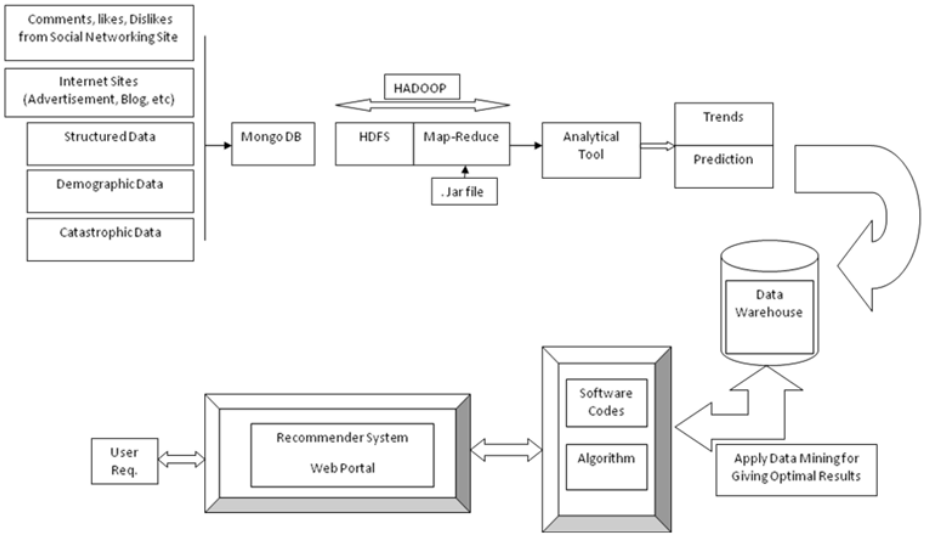
Hadoop Distributed File System: HDFS, the storage layer of Hadoop, is a distributed, scalable, Java-based file system adept at storing large volumes of unstructured data. Map Reduce is a software framework that serves as the compute layer of Hadoop. Map Reduce jobs are divided into two (obviously named) parts. The “Map” function divides a query into multiple parts and processes data at the node level. The “Reduce” function aggregates the results of the “Map” function to determine the “answer” to the query. Hive is a Hadoop-based data warehousing-like framework originally developed by Facebook. It allows users to write queries in a SQL-like language called HiveQL, which are then converted to Map Reduce. This allows SQL programmers with no Map Reduce experience to use the warehouse and makes it easier to integrate with business intelligence and visualization tools such as R studio, Statistica, etc. Pig Latin is a Hadoop-based language developed by Yahoo. It is relatively easy to learn and is adept at very deep, very long data pipelines (a limitation of SQL.) HBase is a non-relational database that allows for low-latency, quick lookups in Hadoop. It adds transactional capabilities to Hadoop, allowing users to conduct updates, inserts and deletes. EBay and Facebook use HBase heavily. Flume is a framework for populating Hadoop with data. Oozie is a workflow processing system that lets users define a series of jobs written in multiple languages – such as Map Reduce, Pig and Hive -- then intelligently link them to one another. Oozie allows users to specify, for example, that a particular query is only to be initiated after specified previous jobs on which it relies for data are completed. Parallel database systems and Map Reduce systems (most notably Hadoop) are essential components of today's infrastructure for Big Data analytics. These systems process multiple concurrent workloads consisting of complex user requests, where each request is associated with an (explicit or implicit) service level objective.

V Proposed Work

We know unstructured data is one without a defined data model or cannot be easily usable by a computer program. In a structured document, certain information always appears in the same location on the page. Big data is now a big problem as the volume; variety and velocity of data coming into enterprises continue to reach extraordinary levels. This unexpected growth means that not only must you understand big data in order to process the information that truly counts, but you also must analyze the possibilities of what you can do with big data using big data analytics. Big data analytics is the process of analyzing big data to find hidden patterns, unknown correlations and other useful information that can be extracted to make better decisions. With big data analytics, scientists and others can analyze huge

volumes of data that old analytics and business intelligence solutions can't find. Using high-performance data mining, predictive analytics, text mining, forecasting and optimization on big data, perfect innovation can be extracted to make the best possible decisions. Big data analytics is helpful in managing more or diverse data. It also helps to generalize new questions from observation, formulating new hypotheses, explore and discovery of new processed concepts, and making decisions from testing. The main efforts done by big data analytic is the use of new analytics techniques on either new data or data that has been mixed in new ways. Big data Analytics used the tool Hadoop for processing the unstructured data. The main point is whether Hadoop will become as indispensable as database management systems. Data analysis is a do-or-die requirement for today's businesses. Today, only 8 percent of large global organizations are using big data analytics to identify patterns [15]. In proposed system (Fig: 3) data from different location whether it is structured, semi-structured or unstructured are collected to a single point and with the help of Hadoop ecosystem we store the data at a single point in HDFS. With the help of inbuilt function map-reduce of hadoop we will reduce the data in relevant format. Analytical tool R studio and Statistics will be used to get the predictive results, different patterns from the data, which will help to make better business decision. The recommender system will give results on the basis of analyzed data which we got as an output from the first part. As per the user request or demand it will apply collaborative algorithm and different mining technique to show the result. The recommender system will able to:

- Monitor trends for home, prices, and sales.
- Do detailed site analysis using demographic data.
- Provide clients with customized market reports.
- Create a commercial real estate analysis using cap rate and ROI.
- Valuation analysis.



(Fig: 3 proposed model)

VI Conclusion & Future Work

In this paper, we propose the use of Big Data Analytics for analyzing the real estate enterprise data, which help to:

- Refine sales and marketing.
- Upgrade asset management
- Sharpen strategic location determination
- Improve fraud detection and preventation

We discussed a framework based on Hadoop for Big Data Analytics. We can manage the Big Data characteristics of large volumes of enterprise data. If enterprise has an unmet business need for strategic decision making with a high degree of processing, a Revolution Analytics and Hadoop combination offers significant opportunity to gain advantage in real estate.

VII References

[1] Big data Analytics for Dummies by Michael wessler, OCP& CISSp.

[2] Big data definition by IBM.

[3] Apache Hadoop Project http://hadoop.apache.org/

[4] R. Magoulas and B. Lorica,"Introduction to Big-Data", Release 2.0(Sebastopol O, Reilly Media), Feb, 2009.

[5] Suning T. Embrace the Era of Big Data [J]. Business Research, 2013 (3): 1-1.

[6] Guojie L, Xueqi C. Research Status and Scientific Thinking of Big data [J]. Bulletin of Chinese Academy of Sciences, 2012, 27(6): 647-657.

[7] Brown B, Chui M, Manyika J. Are You Ready for the Era of 'Big Data' [J]. McKinsey Quarterly, 2011, 4: 24-35.

[8] IBM 2012, what is big data: Bring big data to the enterprise, http://www01.ibm.com/software/Data/bigdata/, IBM.

[9].Amazon, Amazon Elastic Map Reduce (Amazon EMR),http://ieeexplore.ieee.org/xpl/articleDetails.jsp.

[10.] J. Dean and S. Ghemawat, Map Reduce: Simplified Data Processing on Large Clusters, In Proceedings of the OSDI'04: Sixth Symposium on Operating System Design and Implementation
(2004) 137-150.

[11.] J. Dean and S. Ghemawat, Map Reduce: A Flexible Data Processing Tool, Communications of the ACM 53 (2010) 72-77.

[12.] A. Thusoo, J. Sen Sarma, N. Jain, Z Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, Hive: a warehousing solution over a map-reduce framework, Proceedings of the VLDB Endowment 2 (2009) 1626-1629.

[13.] K. Shvachko, H. Kuang, S. Radia, R. Chansler, the Hadoop Distributed File System, In Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies (2010) 1-10.

[14]ApacheHadoopProject http://hadoop.apache.org/

[15] Larry Barrett, "Big data analytics: the enterprise's next great security weapon?" February 2014.

A Study of Component Based Software Engineering: Technologies, People Issues, and New Challenges For Software Industry

Shabnam Vashistha
Shabnam01@rediffmil.com

A B S T R A C T

Component-based software development is the most promising way associated system building by plugging together all useful components. Most software systems are variants of systems that had been already developed. Hence new systems can be developed partially if not completely from the pre-existing systems by reusing it. This brings the great idea of reusability and gave a nice concept of Component Based Software Development. Component-based software development approach is focused on the idea to develop software systems by selecting components and then to assemble them with a well-defined software architecture design. The new software development paradigm is much different from the traditional one. We have discussed on current component-based software technologies, describe their advantages and disadvantages, and discuss the features they inherit. We also discuss people issues about component-based software. The objective of this paper is to gain attention towards new challenges and tries to answer the question of new challenges of component based software engineering for software industry.

1. Introduction

In early days, software engineering approach was not very good. In 1970s, structured programming gave a little shift in software engineering. Then around 1980s object oriented programming with some advancement explores new areas in software engineering but it was not enough to cope with rapidly changing requirements of present day's applications. With introduction of Component Based Software Development (CBSD), the industry is moving in a new direction and in new era. Component based systems are easier to assemble and therefore less costly to build than developing such systems from scratch. The importance of component based development lies in its efficiency and flexibility. In addition, CBSE encourages the use of predictable architectural patterns and standard software infrastructure, thereby leading to a higher result.

1.1 Software Architecture. This new component based software development approach is very different from the traditional approach in which software systems can only be implemented from existing scratch. These reusable commercial components can be developed by different developers using different languages and packages and on different platforms.

The life cycle and software engineering model of CBSD is much different from that of the traditional ones.

There are no predefined or existing standards or guidelines in this new area, and we do not even have a unified definition for "component".

Component

A Component is an independent reusable software unit which be assembled with others component for making software.

In general a component has three main features:

1) A component is an independent, replaceable software part of a system that fulfills a clear function.
2) A component works in the context of a well defined architecture.
3) A component communicates with other components by its interfaces.

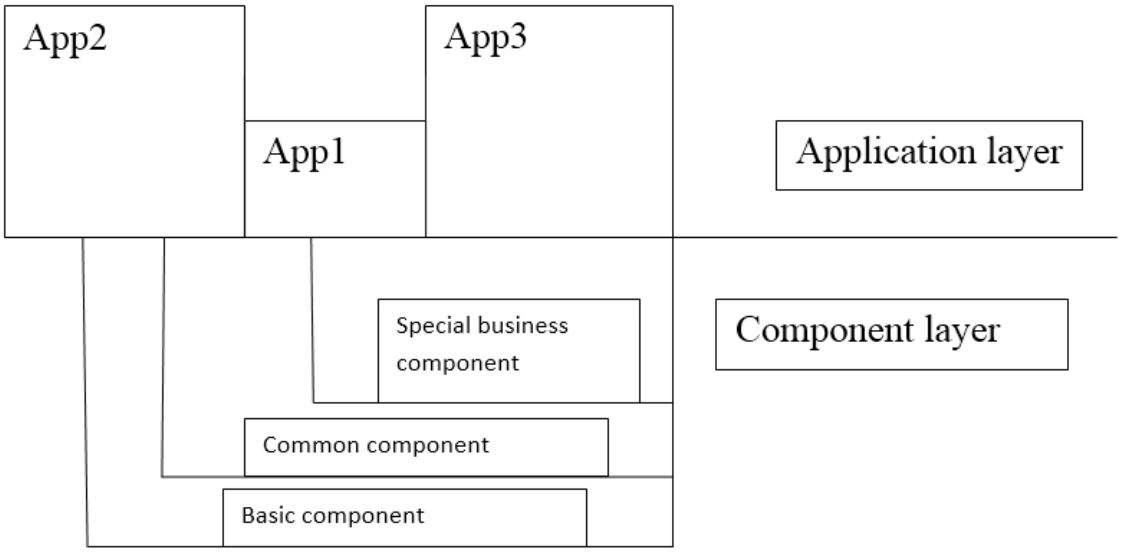


Figure 1. System architecture of component-based software systems

Layer first is the application systems supporting a business. The second layer consists of components engaged in only a specific business or application domain. The third layer is cross-business middleware components

Consisting of common software and interfaces to other established entities. Finally, the lowest layer of system software components includes basic components that interface with the underlying operating systems and hardware.

2. Current Component Technologies

There are many technologies which are used for component development, such as COM (Component object model), Visual Basic Controls (VBX), class libraries and Java Beans and some languages such as Visual Basic, C++, Java, and the supporting tools to share and distribute application pieces. But all of these approaches rely on certain underlying platform and services to provide the communication and coordination necessary for the application. Among the component infrastructure technologies some have been extended somewhat standardized: OMG's CORBA, Microsoft's Component Object Model (COM) and Distributed COM (DCOM), and Sun's Java Beans and Enterprise JavaBeans. (2.1) CORBA is an open standard for application interoperability that is defined and supported by the Object Management Group (OMG). It define component interoperability and allows applications to communicate with one another. It offers a consistent distributed programming and better run-time environment over common programming languages, operating systems, and distributed networks. (2.2) COM defines how components and their clients interact. This interaction is defined such that the client and the component can connect without the need of any Intermediate system component. This enables on-line software update and cross-language software reuse.

DCOM is designed for use across multiple networks transports Sun's Java-based component model consists of two parts:

(1) The JavaBeans for client-side component development
(2) Enterprise JavaBeans (EJB) for the server-side component development. The JavaBeans Component architecture supports applications of multiple platforms, as well as reusable, client-side and server-side components.

Java platform offers an efficient solution to the portability and security problems through the use of portable Java byte codes and the concept of trusted and entrusted Java applets. Java provides a universal integration and enabling technology for enterprise application development, including

1) Inter operating across multi vendor servers;
2) Propagating transaction and security contexts;
3) Servicing multilingual clients
4) Supporting ActiveX via DCOM /CORBA bridges.

JavaBeans and EJB extend all native strengths of Java including portability and security into the area of component-based development.

3 People Issues

This research paper discusses some human, social, non technical and organizational issues affecting Component based software engineering (CBSE) processes and design in organizations. Software systems do not exist in isolation they are used in social and organizational contexts. Experience and many studies show that the major cause of most software failures is the people behavior rather than technical issues. Even with the availability of a wide range of advanced software development methodologies, techniques and tools, serious problems with software are still being faced. It is the people and culture of the organization that determines how any system is used. For example poor training may result in people not co-operating with the information system leading to failure and project abandonment .**Curtis et al** has highlighted that human, social and organizational considerations affect software processes and introduction of software technology. **Le Quesne,** agreed that certain aspects of the design of information systems would make its likely success dependent on characteristics of the particular organization environment.

Social and organizational factors Studies by **Le Quesne, Grudin** and others have identified a number of social and organization issues that affect the software systems. Incremental and evolutionary are two examples of phased development strategy. The advantages are that the selected portion is delivered early, increased user acceptance and adaptation.

4 New Challenges

Software is no longer marginal in technical systems but central factor in many areas. We are witnessing an enormous expansion in the use of software in business, industry, administration and research.

System features mainly based on software functionality rather than characteristics .for example in car industry, the service sector and in schools increasing numbers of software users are non-experts. These trends place new demands on software usability, robustness; simple installation and integration become the most important features of software.

As software utilization has increased in wider area, the demand for the integration of different areas has increased. There are two integration

(a) Vertical integration: - data and processes at different levels are integrated.
(b) Horizontal integration: - similar types of data and processes from different domains are integrated.

For example, in industrial process automation system, at the lowest levels of management (Field Management level), the data collected from the process and controlled directly, then is further processed for analysis and combination with the data provided from the market and finally published on the Web .software development addressed many challenges of increasing complexity and dependence on delivery deadlines while ignoring the evolutionary needs of the system. This has led to a number of problems: failure of the majority of projects to meet their deadline, budget, and quality requirements assessment and the continued increase in the costs associated.

To meet these challenges, software development process must be able to cope with complexity and to adapt changes made in industry. If new software products are each time to be developed from Scratch, these goals cannot be achieved.

The key solution for this problem is reusability of system component. From this perspective Component-based Development (CBD) appears to be the right approach. In CBD software systems are built by assembling components already developed and prepared for integration. CBD has many advantages. These include more effective management of complexity, reduced marketing time, increased productivity, improved quality, greater degree of consistency, and wider range of usability

However, there are several challenges which are industry is facing.

4.1 Unclear and ambiguous requirements.

Requirements management is an important part of the development process and its main objective is to define consistent and complete component requirements. Reusable components are to be used in different applications, some of which may yet be unknown and the requirements of which cannot be predicted. This applies to both functional and non-functional requirements.

4.2 Time required for the development of components.

This factor for reusable components are the increased time and effort required, the building of a reusable unit requires three to five times the effort required to develop a unit for one specific purpose.

4.3 Usability and reusability - a conflict.

A component should be sufficiently general and adaptable and therefore will be more complex. A requirement for reusability may lead to another development approach, for example building a new, more abstract level which gives less flexibility and fine tuning, but achieves better simplicity.

4.4 Costs for Component maintenance.

When application maintenance costs can decrease, component maintenance costs can be very high since the component must respond to the different requirements with different reliability requirements and perhaps requiring a different level of maintenance support.

4.5 Reliability to changes.

Components and applications have separate life cycles for their requirements, so there is some risk that a component will not completely satisfy the application requirements or it may include concealed characteristics that is not known not known to application developers. When introducing changes on the application level (updating of operating system, or changes in the application), there is a risk that this will cause system failure. To enjoy the advantages and avoid the problems and risks, we need a systematic approach to component-based development at the process and technology levels.
Overviews of certain CBSE disciplines and some of the relevant trends and challenges in the near future are presented below.

4.5.1 Component Specification

For a common understanding of component based development, the starting point is an agreement of what a component is and what it is not.
As a generic term the concept is pretty clear a component is a part of something but this is too vague to be useful. The definition of a component has been widely discussed.

However, we shall adopt Szyperski's definition which is the most frequently used today:

A software component is a unit of composition with contractually specified interface and explicit context dependencies only.

A software Component can be deployed independently and is subject to composition by third parts.
Component Interface, Component Implementation
There are also different comprehension of Component Based Development in industry and academia. While researchers in academia define components as well defined entities (often small understood functional and non-functional), industry sees components as parts of a system which can be reused, but

are not necessarily. A component can be an important part of a system the adaptation of which may require much effort. Such components are of extreme importance as well, as the larger the components, the greater the productivity level which can be achieved by their reuse.

5. Conclusion and Future Work

In this paper, we survey current component-based software technologies and the features they inherit. We have discussed on some people issues.

The progress of software development in the near future will depend very much on the successful establishment of CBSE and this is recognized by both industry and academia. In this paper our focus will be on new challenges faced by component based software engineering and to find out their solutions.

6. References

[1] A.W.Brown, K.C. Wallnau, "The Current State of CBSE," IEEE Software, Volume: 15 5, Sept.-Oct. 1998, pp. 37 – 46.

[2] M. L. Griss, "Software Reuse Architecture, Process, and Organization for Business Success," Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering, 1997, pp. 86-98.

[3] P.Herzum, O.Slims, "Business Component Factory – A Comprehensive Overview of Component-Based Development for the Enterprise," OMG Press, 2000.

[4] Hong Kong Productivity Council, http://www.hkpc.org/itd/servic11.htm, April, 2000.

[5] IBM:http://www4.ibm.com/software/ad/sanfrancisco, Mar, 2000.

[6] I.Jacobson, M. Christerson, P.Jonsson, G. Overgaard, "Object-Oriented Software Engineering: A Use Case Driven Approach," Addison-Wesley Publishing Company, 1992.

[7] W. Kozaczynski, G. Booch, "Component-Based Software Engineering," IEEE Software Volume: 155, Sept.-Oct. 1998,

[8] M.R.Lyu (ed.), Handbook of Software Reliability Engineering, McGraw-Hill, New York, 1996.

[9] Microsoft:http://www.microsoft.com/isapi, Mar, 2000.

[10] J.Q. Ning, K. Miriyala, W. Kozaczynski,, "An Architecture-Driven, Business-Specific, and Component Based Approach to Software Engineering," Proceedings Third International Conference on Software Reuse: Advances in Software Reusability, 1994, pp. 84 -93.

[11] OMG: http://www.omg.org/corba/whatiscorba.html, Mar, 2000.

[12] G. Pour, "Component-Based Software Development Approach: New Opportunities and Challenges," Proceedings Technology of Object-Oriented Languages, 1998. TOOLS 26., pp. 375-383.

[13] G. Pour, "Enterprise JavaBeans, JavaBeans & XML Expanding the Possibilities for Web-Based Enterprise Application Development," Proceedings Technology of Object-Oriented Languages and Systems, 1999, TOOLS 31, pp.282-291.

[14] G.Pour, M. Griss, J. Favaro, "Making the Transition to Component-Based Enterprise Software Development: Overcoming the Obstacles – Patterns for Success," Proceedings of Technology of Object-Oriented Languages and systems, 1999, pp.419 – 419.

[15] G.Pour, "Software Component Technologies: JavaBeans and ActiveX," Proceedings of Technology of Object-Oriented Languages and systems, 1999, pp. 398 – 398.

[16] C. Rajaraman, M.R. Lyu," Reliability and Maintainability Related Software Coupling Metrics in C++ Programs," Proceedings 3rd IEEE International Symposium on Software Reliability Engineering (ISSRE'92), 1992, pp. 303-311.

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

-

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Address of the Editorial Office:

Enriched Publications Pvt. Ltd.
S-9,IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR -5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-45525005