IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010
ISSN (Online): 1694-0814
www.IJCSI.org

Reengineering Of Education Institute And Coupling Measurement

Dr Ashok Kumar [1], Anil Kumar[2], Kulvinder Singh[3]

1.Professor, Dept. of Comp. Sc. & App. Kurukshetra University, Kurukshetra, India

2. Asst. Professor, Comp. Sc. & Engg. Vaish College of Engineering, Rohtak, India.

3. Asst. Professor, Comp. Sc. & Engg. Vaish College of Engineering, Rohtak India.

Abstract

This paper is based on reengineering of Education institutes[1,2] in such a way that coupling risk should be less as compared to existing systems. Here, we will measure the complexity (based on coupling factor) of modules during reengineering of modules design. As we know that the coupling[3,4] is one of the properties with most influence on maintenance, as it has a direct effect on maintainability.

In general while any module is reengineered [5], one of the goals of OO software designers is to keep the coupling of system as low as possible. Classes of the system that are strongly coupled are most likely to be affected by changes and bugs from other classes. As the coupling between the classes of the system is increased, it result in increased error density.

The work described in this paper measure the coupling not only through classes of the system, but also through the Packages[6] that are included during reengineering of the module design.

Coupling between packages is the manner and degree of interdependence between them. Theoretically, every package is a stand-alone unit, but in reality packages depend on many other packages as either they require services from other packages or provide services to other packages. Thus, coupling between packages cannot be completely avoided but can only be controlled. The coupling between packages is an important factor that effects the quality or other external attributes of software, e.g., reliability, maintainability, reusability, fault-tolerance etc.

In this paper, some measures are proposed for measurement of coupling at the package-level[7] in order to achieve good quality software systems.

In this paper, we propose metrics[8] for measuring the coupling between packages in a software system.

Keyword

Education Institute, Coupling, Package, Metrics ,System Representation and Definition

Introduction

Due to increasing demand of software maintenance, today reengineering techniques is one of the best choice, to full-fill the requirement of the public sector/private sector. While reengineering takes place in existing system, the complexity of modules is determined in early stage (at design time), due to this, software designer, easily determine, about resources that is required during reengineering of any project.

This paper is based on the measurement of complexity of modules, by using different types of coupling metrics, at design time.

The backbone of any software system is its design. It is skeleton where the flash (code) will be supported. And while determining the complexity[9] of the modules, here will use OO paradigm, which is very popular concepts in today's software development environment. They are often heralded as the "silver bullet" for solving software problem.

Literature Overview

Hamper and Champy[10] (1993) defines the business process reengineering as " the fundamental rethinking and radical redesign of the business processes to achieve dramatic improvement in critical, contemporary measures of the performance, such as cost, quality, service , and speed". Business processes and redesigns to improve performance.

The importance of business processes reengineering (BPR) is crucial for public sector organizations (Parys and Thijs, 2003). It is difficult to deploy BPR efforts in public organizations [11] (Parys and Thijs, 2003; Robert, 1994; Thong, Yap, and Seah, 2000)[12]. The government organizations re usually attached to many other departments and ministries. The change in one unit requires change in other interlinked organizations. It is therefore necessary to handle all these problems to successfully implement the BPR strategy in the public sector organizations (Parys and Thijs, 2003).

Tenner and Dectoro[13] (1992) defines process as a single or combination of tasks that add value to inputs to convert them into outputs by the application of human interaction, methodologies and techniques. The author confines the key points and describes stages to improve business processes using the step by step procedure to achieve real performance goal. Individuals at any level in manufacturing, service or the public sector can benefit from this approach which enhances the chances of success in improving organization wide performance. It is designed for leaders at any level who are committed to drastically improving their organization's performance through redesigning its processes.

The business process consists of different activities which defines the pattern of work in the organizations (Sethi and King, 2003)[14]. The efficient processes serve to satisfy the customers by converting input resources to desired output (Field, 2007; Hammer and champy, 193; Harrison and Pratt, 1993; Snee, 1993)[15].

Evans[16] (1993) signifies the importance of analyzing the existing business processes and the organizations to identify bottlenecks in the systems. The author translates this phase 'As Is' step of BPR. The other important phase is ' To Be' which describes the desired performance achievement level of the business process. BPR attempts to fill the gap between these two organizational situations. Business process analysis attempts to achieve operational efficiency by reducing time and cost factors (Cook, 1996; Davenport, 1993; Day, 1994; Roy, 2005; Wing and Ahmed, 2003; Muthu, Whitman and Cheraghi, 1999).

Fitzgerald and Murphy (1996) suggest four crucial phases for successfully implementing the BPR strategy in the organizations. First, the core business processes to be redesigned should be selected. Second, the process team should be established to reengineer the core business processes. Third, the current processes may be analyzed and examined to find out bottlenecks in the systems. This phase also determines the satisfaction level of stakeholders with the process outcomes. The last phase encompasses the strategy to reengineer the process to improve performance.

Problem Description

While we reengineering the education institutes, there are 'n' number of packages, the 'n' number of packages has 'n' number of classes, the 'n' number of classes has 'n' number of methods, and the 'n' number of method has 'n' number of operation to perform their activities, these are depends upon other methods, other classes and may or may not depends upon other packages to perform their task.

System Representation and Definition

Definition 1: (*System, Packages)*[17,18]

We consider Modified OO System as a set of packages $P = \{ p_1, p_2, \ldots\ldots p_n\}$. The number of packages in the system P is $n = | P |$

Definition 2: (*Clases of a package)*[19]

A package has a set of classes, and a class has a set of methods. For each package p E P. M (p) = $\{ m_1, m_2, \ldots\ldots, m_n\}$ represent its set of methods in the system is defined as M(P).

Definition 3: *(Conceptual similarities between method)*

The conceptual similarities between method m_k E M(P) and m_j = M(P), CSM(m_k,m_j), is computed as the cosine between the vectors Vm_k and Vm_j, corresponding to m_k & m_j in the semantic space constructed.

Definition 4: (*conceptual similarities between a package's of classes and methods)*[20]

Let p_k E P and p_j E P are two distinct (p_k *not equal to* p_j) packages in the system. Each packages has a set of classes and each set of class has a set of method $M(p_k) = \{ m_{k1}, m_{k2},\ldots,m_{kr}\}$, where r = | $M(p_k)$ | and $M(p_j) = \{ m_{j1}, mj_2,\ldots,mjt\}$, where t = | $M(p_j)$ |

Between every pair of methods (M_k,M_j), there is similarities measure, conceptual similarities between package's of the class and between method's of class(CSPCM(M_k,M_j))

Which is the average of conceptual similarities between method M_k and all the method from package of class C_j.

Definition 5:*(Conceptual similarities between two package)*

We define the conceptual similarities between two package(CSBP) p_k E P and pj E P *is* *CSBP(*p_k, p_k) = $\sum^{r}_{a=1}$ CSPCM(M_{ka},p_j))

Which is the average of the similarities measures between all unordered pairs of methods from package p_k and p_j.

The definition ensures that the conceptual similarities between two packages is symmetrical as CSBP(p_k ,p_j) = CSBP(pj ,pk)

Proposed work

Here, we will specify metrics which is very beneficial to measure the coupling during reengineering of the educational institutes. These metrics are given below:

Response For Packages (RFP)

Response for package measure the complexity of the package in terms of classes in the packages plus methods in the classes. Not include inherited packages, but included inherited classes in the specific packages.

Message Passing among packages (MPP)

This metrics measures the number of message passing among objects of packages. A larger number indicates increased coupling between this package and other packages in the systems. This makes the package more dependent on each other, which will increases the overall complexity of the system and make packages more difficult to change in future.

Coupling between Objects(CBO) (Chidamber & Kemerer) [21]

CBO is the total of the number of classes that a class referenced plus the number of classes that referenced classes. It was only counted once.

Fan-out

Fan-out is defined as the number of other classes referenced by a class in the package.

Fan-in

Fan-in is the number of other packages that references to a package.

Efferent Coupling

Efferent coupling is viewed as equivalent to Fan-Out

Afferent Coupling

Afferent coupling is viewed as equivalent to Fan-In

There are number of other useful metrics at package level[22]:

a. **Reuse Ratio** is calculated as

$$\text{Reuse ratio} = \frac{\text{Number of super package above this package in the Package hierarchy}}{\text{Total number of packages in package hierarchy}}$$

b. **Specialization Ratio** is calculated as

$$\text{Specialization ratio} = \frac{\text{Number of sub package below this package in the package hierarchy}}{\text{Number of super package above this package in the Package hierarchy}}$$

c. Number of instance variable, number of modifiers, number of interfaces implemented and number of package imported :

These gives additional information about the class's level in the package level of semantic complexity. As with methods, large values for these can suggest that a class doing too much.

Related Works

For OO Systems, most of the coupling metrics, exists for measurement of coupling at the higher levels of abstraction in OO systems. Other, works related to the packages or other higher abstraction level has been carried out in.

Conclusion

The goal of this paper, provided certain types of package level metrics, that is used to determine coupling of modules during reengineering of modules design, and help the designer to reduces the coupling among modules, due to this increasing user productivity and scalability, improve vendor independence, enhance scalability, increase manageability and more. As we know that, due to the productivity, it will create in the more economic value for each unit of cost. This is different from cost reduction. It specifically defines different types of sub-modules complexity and once the coupling is clearly determined, then it is very beneficial in future requirement of reengineering of modules, because requirement obviously changes, after a specific time period. And whenever needed, further to reengineering the modules it is easily takes place.

References

1. "The National Information Infrastructure Agenda for Action"
2. "Perspective on the national Information Infrastructure"
3. F. Pilat, W. Fischer, S. Peggs, V. Ptitsyn, S. Tepikian, .The Coupling Correction System at RHIC: Results for the Run 2000 and Plans for 2001., PAC 2001
4. V.Ptitsyn, J. Cardona, J-P. Koutchouk, F. Pilat, .Measurement and Correction of Linear Effects in the RHIC Interaction Regions., PAC 2001
5. Booch G. Object Oriented Analysis and Design with Appli-
6. cations. 2nd Ed., Redwood City: Benjamin Cummings Pub-lishing Company Inc., LA, USA, 1994
7. Booch G. Object Oriented Analysis and Design with Appli-
8. cations. 2nd Ed., Redwood City: Benjamin Cummings Pub-lishing Company Inc., LA, USA, 1994.
9. Eder J, Kappel G, Schre° M. Coupling and cohesion in object-oriented systems. Technical Report, University of lagenfurt,1994.
10. Hitz M, Montazeri B. Measuring coupling in object-oriented systems. Object Currents, 1996, 1(4): 124{136.
11. Hammer, M. and Champy, j., 1993. "Reengineering the Corporation : a manifesto for business revolution", HarperCollins Publishers, Inc. New York
12. Rober, L. (1994). "Process Reengineering: the key to achieving breakthrough success", Milwaukee, Wisconsin, ASQC Quality Press.
13. Thong, J., Yap, C.S. and Seah, K.L., 2000. "Business Process Reengineering in the public Sector: the case of the Housing Development Board in Singapore", Journal of Management Information System, 17(1), pp. 245-271
14. Tenner, A.R., Dectoro, I.J.,1992. "The Process redesign – the implementation guide for manager", Prentice Hall, New Jersey.
15. Sethi, V. and King, W.R., 2003. "Organizational transformation through business process reengineering: applying the lessons learned", Pearson education, Inc, Singapore
16. Harrison, D.B. and Pratt, M.D., 1993. " A Methodology for reengineering business", Planning Review, 21(2), pp. 6-11
17. Evans, K., 1993. "Reengineering and Cybernetics", American Programmer, 6(11), pp. 10-16
18. Abreu, F. B., Pereira, G., and Sousa, P., "A Coupling-
19. Guided Cluster Analysis Approach to Reengineer the Modularity of Object-Oriented Systems", in Proc. of Conf. on Software Maintenance and Reengineering (CSMR'00), Zurich, Switzerland, Feb. 29 - Mar. 3 2000, pp. 13-22.
20. Arisholm, E., Briand, L. C., and Foyen, A., "Dynamic coupling measurement for object-oriented software", IEEE Transactions on Software Engineering, vol. 30, no. 8, August 2004, pp. 491-506

21. riand, L., Wust, J., and Louinis, H., "Using Coupling Measurement for Impact Analysis in Object-Oriented Systems", in Proc. of IEEE International Conf. on Software Maintenance, Aug. 30 - Sept. 03 1999, pp. 475-482.
22. Briand, L. C., Devanbu, P., and Melo, W. L., "An investigation into coupling measures for C++", in Proc. Of International Conference on Software engineering (ICSE'97), Boston, MA, May 17-23 1997, pp. 412 - 421.
23. Chidamber, S. R. and Kemerer, C. F., "Towards a Metrics Suite for Object Oriented Design", in Proceedings of OOPSLA'91, 1991, pp. 197-211
24. D.L. Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules," Communications of the ACM, Vol. 15, No. 12, Dec. 1972, pp 1053-1058.

Dr. Ashok Kumar received his Ph.D degree from Agra University, Agra, India. He joined Kurukshetra University, June 1982. He is working as Professor in the Department of Computer Science & Application, Kurukshetra University, Kuruks hetra – 1361199 (Haryana), India, since June 1982. He was awarded four Ph.D award from Kurukshetra University. He has published more than 60 national and international papers. He has attended more than 30 national and international seminars. His area of interest are software engineering, operational research, networking and operating system.

Anil Kumar received his bachelor degree from Delhi University, Delhi, India and Master degree from IGNOU, India and M.Tech in Computer Science & Eng. From Kurukshetra University, Kurukshetra, India in year 2002 and 2006. Currently he is pursuing Ph.D in Computer Science from the Department of Computer Science & Application – Kurukshetra University, Kurukshetra, India. Currently is Asst. Professor in Computer Science & Engineering Department in Vaish Engineering College, Rohtak, Haryana, India since September, 2006. He had also worked in software industries more than three years & being a lecture in other engineering college more than two years. His research areas include Software engineering, Reengineering, Software Metrics, Object Oriented analysis and design, Reusability, Reliability.

Kulvinder Singh received the M.Tech.(CSE) degree in 2006 and the M.Phil.(CS) degree in 2008 from Ch. Devi Lal University Sirsa(Haryana), India. At present he is working as a Assistant Professor in Vaish College of Engineering, Rohtak, India. He is a member of IEC. He presents many research papers in national and international conferences. His interest areas are Networking, Web Security, Internet Congestion and Fuzzy Database.