Critical Parameters for Coupling and Cohesion of S/W Reusability Problem Domain

A. Ravi, Nirmala

Abstract- Interdependency between different s/w modules that are linked in specific application areas and/or dependent functions within single module, carry risks while they are adapted for s/w reusability. Coupling and cohesion are related terms that contribute a lot in s/w reusability. While well structured programs are risk free when used for specific application, huge s/w when developed for similar applications, then reusability of part of or full modules cause risks. Features like hardware capacity, operating system, file structure, network capability, interoperability, scalability and security (popularly known as 'ilities') are parameters that put in influences on the coupling and cohesion. This paper attempts to determine criticality of some of the parameters so as they form critical elements causing the risks on coupling and cohesion. Even though the paper does not present optimization techniques to consider these parameters for s/w reuse, the parametric study results will be of immense use to s/w reusability for obtaining optimum solutions. Experiments with four s/w modules written in Java have been carried out with different entities that form different coupling and cohesions. Observance from the results has yielded to identifying critical elements.

Keywords: s/e reusability, coupling and cohesion, critical parameters

I. INTRODUCTION

Coupling is the degree of interdependency of a program module that relies with other program modules. But cohesion on the other hand relies on dependency of functions or sub modules within one module itself. It is hence logical, when a single module becomes large in size, low cohesion of it will create problems. Coupling is usually contrasted with cohesion and low coupling is often correlates with high cohesion [1]. Therefore low coupling is habitually a sign of well structured system, when combined with high cohesion. But now-a-days reuse of s/w, thus mounting coupling, has become increasingly common in industries. As risks involved in s/w reusability is proportional to the volume of coupling, critical entities related to coupling need to be studied.

There are two domains that contribute to dependencies as well as interdependencies of large s/w modules. These two domains are solution domain and problem domain. While solution domain mostly bothers about technical features like hardware capacity, operating system, file structure, network capability, interoperability, scalability and security (popularly known as 'ilities'), the problem domain on the other hand bothers mostly about user - requirements like usability [2]. In principle, user domain is either satisfied or not satisfied, but difficult to represent anything in between these two extremes.

Manuscript Received November, 2013.

A. Ravi, Research scholar in M.S. University, Chennai, India.
Dr.Nirmala, M.C.A.,Ph.D.,Associate Professor Department Of Computer Science Quaid-E-Millath Govt.College For Women Chennai, India.

Therefore user domain is functional in that aspect, and it is hence not exactly measurable. But solution domain is non-functional and it is measurable. We have made an attempt to combine both these domains as problem domain and delimited our scope of work to certain chosen parameters for the purpose of analytical study. The chosen parameters contribute to a great deal on coupling and cohesion. We have performed a series of parametric study that would help in identifying critical parameters that would influence coupling and cohesion. Each parameter is translated into some measurable property of the final property. In software engineering, it is essential to determine the most critical non-functional parameters that can dominate the best choice of development technique and internal design. For example if reliability becomes critical, bug detection could be made with peer reviews. Therefore analyzing these parameters for the purpose of determining critical entities becomes necessary. In view of this, we have proposed to experiment with four software programs written in Java for few possible re-usable application areas have been tried out and the coupling are calculated on the basis of certain chosen parameters. As we are only interested in identifying critical parameters, we have not presented the details of s/w modules and the application areas, but only the results from the experimental runs are considered for analytical study.

Apart from coupling values, cohesion values are also calculated based on a single chosen application program. Comparisons between these two are made out of these experiments and important conclusions are drawn on the critical parameters.

II. NON-FUNCTIONAL PARAMETERS AND METRICS

Non Functional Requirements (NFR) are mostly expressed in terms of measurable entities such as 'usability' that is a subjective term; 'security' and 'reliability' which are quality terms, and 'interoperability' and 'scalability' which are performing terms. In addition, NFR also involve: Time to implement s/w as a product or project, cost: which is related to productivity, CPU Speed, memory and secondary storage capacities that are considered to be utility terms [3]. **A**ll the above entity requirements can be specified in measurable values. The actual problem lies in the way of measuring these. For example, an end user may ask for something that is "quality higher than the previous product", without really quantitatively specifying how that quality is to be measured. It could be in the form of 'maintenance cost' or time caused due to failures, repairing efforts, fixing time, etc. Perhaps the end user could say he wanted the new product that needs to be "smarter" than the old. Old product could be compared with a new product so that the decision capabilities can be put into the new product. Or the end user could ask for easy to use the new product, than the old one [6]. All these requirements that seem to be qualitative instead of quantitative, can still be expressed in terms of hard numbers if the end user and the developers are willing to work together to establish measurements. But how to make sure that the end user's requirements are incorporated everything he had thought in terms of characteristic values? At the same time, a very small piece of the desired quality cannot be stated as some objective characteristic that can be quantified. But however we can assign measurements to that quality. This can be achieved by analyzing such terms of metrics that becomes critical for a parameter [3].

2.1 Critical S/W Reusable Metrics

In S/W reusable problem domain, the reusable components such as functions and packages produce various metrics for program modules. These metrics reduce risk factors involved and improve the level of reusability. Four chosen broader metrics have been considered and listed, and explained. They are: 1. Productivity enhancement; 2. Faster deployment; 3. Finalization of product and 4. Quality enhancement. Productivity of any s/w product mainly depends upon cost and time taken from development stage to implementation stage. Reduction in many entities such as manpower during the development stage of the product could eventually decrease the product cost. Meticulous planning for reusing certain s/w components might decrease the cost. Faster deployment of the product, if so required by the end-user and if that can be determined in advance, the delivery time would be considerably reduced. Finalization of product means ascertaining the exact outcome of the end product that would be presented to the end-user. This would reduce the efforts taken for re-designing and correcting the end product. Quality enhancement needs verification and validation of the working principle of the product that must be carried out along with the design and development of the software product [5]. This should enhance the software quality. Reliability measurement should be carried out every now and then, through analyzing for critical parameter. This will produce fault free software product in most of the cases. But do these metrics rely much in coupling and cohesion?

III. COUPLING AND COHESION IN S/W REUSABILITY

Coupling and cohesion are popular in software reusability. It is essential to know briefly about different types of coupling, so that critical analysis can be carried out based on what metric is affected [7]. Content coupling or Pathological coupling is one that modifies or relies on the internal workings of another module (e.g., accessing local data of another module). Therefore changing the way the second module produces data (location/type) will lead to changing the dependent module. Common coupling or Global coupling is that, when two modules share the same global data (e.g., a global variable or constant). Changing the shared resource implies changing all the modules using it. External coupling occurs when two modules share an external data, or communication protocol, or device interface.

This basically refers to communication with external tools and devices [4]. Control coupling is that module which controls the flow of another, by passing such information on what to do (e.g., passing a flag). Stamp coupling that shares a composite data structure and uses only a part of it, possibly a different part (e.g., passing a whole record to a function that only needs one field of it). This may lead to changing the way a module reads a record because a field that the module doesn't need has been modified. Data coupling is when modules share data, for example, data or control parameters. Each datum is an elementary piece, and these are the only data shared (e.g., passing an integer to a function that computes a square root). Message coupling is the loosest type of coupling. It can be achieved by state decentralization (as in objects) and component communication is done via parameters or message passing. Modules do not communicate at all with one another is known as 'No coupling'. Subclass Coupling and Temporal coupling are out of scope of criticality. Based on the above distinct definitions, criticality can be analyzed so that risks in s/w reusability can be reduced.

IV. PARAMETRIC STUDY AND DISCUSSIONS

The objective of the parametric study is to obtain critical parameters that influence coupling and cohesion. Four program coding were developed with varying degrees of volumes and number of functions and, also with number of connections with other modules. Coupling and cohesions are calculated with some of these parameters using equation (1). The computed values of coupling for the case with and without data and control parameters are presented in Table 1.0. The computed values of cohesion for the case with and without data and control parameters are presented in Table 2.0.

Table legend:- ID: number of input data parameters; IC**:** number of input control parameters; OD: number of output data parameters; OC: number of output control parameters. For global coupling:- GD: number of global variables used as data; GC: number of global variables used as control. For environmental coupling:- MO: number of modules called (fan-out); MI: number of modules calling the module under consideration (fan-in). Coupling is calculated [1] as:

Coupling (C) = 1 – 1 / (ID + 2*IC + OD + 2*OC + GD + 2*GC + MO + MI) ----- (1)

It is obvious that control parameters are twice as much important as data parameters, and hence the coefficient is twice.

Table 1.0 Parametric Values of Coupling

S. No.	PARAMETERS	Values in Modules			
1	Module No.	I	II	III	IV
2	Module Fan in No. (MI)	1	2	3	4
3	Other Modules Fan out No. (MO)	1	2	3	4
4	No. of Input Data Parameters(ID)	4	3	2	1
5	No. of Output Data Parameters(OD)	4	3	2	1
6	No. of Input Control Parameters (IC)	1	2	3	4
7	No. of Output Control Parameters (OC)	1	1	1	1
8	No. of Global Data Parameters (GD)	2	2	2	2

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

9	No. of Global Control Parameters (GC)	3	3	3	3
10	Coupling with All Parameters	0.955	0.958	0.962	0.964
11	Coupling without Data Parameters	0.917	0.938	0.950	0.958
12	Coupling without Control Parameters	0.941	0.944	0.947	0.950

Table 2.0 Parametric Values of Cohesion

Function No.	Data/ Void Function calls (Fan out)	Control Function calls (Fan out)	No. of Control Para-meters	No. of Independent Data Para-meters	Cohesion with All Para-meters	Cohesion without Data Para-meters	Cohesion without Control Para-meters
1	1	1	1	4	0.888	0.800	0.857
2	2	2	1	4	0.917	0.875	0.900
3	3	3	1	4	0.933	0.909	0.923
4	4	4	1	4	0.944	0.929	0.938

From the output of both the above coupling as well as cohesion, we have termed NFRs in terms of parameters such as Productivity; Utilization; Quality; Performance and Finalization as elaborated earlier. The terms listed as Non-Functional Parameters can now be expressed as metrics for these parameters which are presented in Table 3.0.

Table 3.0 Metrics for Non-Functional Parameters

S.No	Non-Functional Parameters	Metrics (Measurable Entities)	Critical Element(s) of Coupling / Cohesion
1	Finalization	Usability	Input / Output Data of Coupling
2	Quality	Reliability	Input Control of Coupling
3	Performance	Interoperability	Fan out of Coupling
4	Performance	Scalability	Fan in of Coupling
5	Quality	Security	Input Control of Coupling
6	Productivity	Time to market	Coupling & Cohesion Values
7	Productivity	Cost	Coupling & Cohesion Values
8	Utilization	CPU Speed	Input / Output of Coupling & Cohesion
9	Utilization	Memory Capacity	Input / Output of Coupling & Cohesion
10	Utilization	Secondary storage Capacity	Input / Output of Coupling & Cohesion

It is seen from the Tables above, each parameter has a critical element of coupling and cohesion. These quantified terms can be used for computing coupling and cohesion for criticality through parametric study experiments on s/w modules. The critical elements considered for the experiments are input / output data and control for computing coupling and cohesion. They are computed so as to validate the criticality through deliberations. This experiment and study are part of a original research area of the authors.

V. CONCLUSIONS

It is clearly demonstrated that for every non-functional parameter there is a corresponding measurable metric that could be quantified. It is shown that the critical entities could contribute to more than one parameter. These critical elements are important that might contribute in risks caused by coupling and cohesion.

REFERENC

1. Wikipedia, 2012,
2. Pressman, Roger S. Ph.D. (1982). Software Engineering - A Practitioner's Approach – Fourth Edition. ISBN 0-07-052182-4
3. Lawrence Chung and Brian A. Nixon, (1995), "Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach", Proc. of IEEE, 17th International Conference on Systems, Seattle, April 24-28, 1995, pp 25-37.
4. Andrea Capiluppi and Cornelia Boldyreff (2007), 'Coupling Patterns in the Effective Reuse of Open Source Software', Proceedings of International Conference on Software Engineering '07, Vol. 1, p.9.
5. Basili V.R., Bri L.C. and Thomas W.M. (1994), 'Domain Analysis for the Reuse of Software Development Experiences', Experimental Software Engineering Group (ESEG), Vol. 11, pp. 86-95.
6. Fenton Norman and Neil Martin (1999), 'Software Metrics and Risk', Proceeding of second European Software Measurement Conference, pp.1-17.
7. F. Beck, S. Diehl (2011), "On the Congruence of Modularity and Code Coupling", Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (SIGSOFT/FSE '11), Szeged, Hungary, September 2011. doi:10.1145/2025113.2025162