Comprehensive Review Paper on Resource Reusability and Utilization

Er. Iqbaldeep Kaur
Department of Computer Science and Engineering
Chandigarh University, Punjab, India

Rajvir Kaur
Department of Computer Science and Engineering
Chandigarh University, Punjab, India

Abstract- Reusability is used to reduce the development time by reusing the already developed components and available information. Over the past decade the concept of reusability has been used to solve problems in different areas. In software it is common to reuse software components. Apart from this, reusability can be used in commercial and professional areas like bank loan process. The bank loan process is a very risky business. Some customers tend to fail most of the times to repay their debt. Hence, previous identification of these potential customers becomes mandatory. And their information is important from the point of view of reusability. This paper overviews the software reusability in bank loan process.

Keywords –Reusability, Software development, bank loan process, Genetic Algorithms, Ant Colony Optimization.

I. INTRODUCTION

Programming Development methodology contains different stages amid the advancement of a product substance. In part based frameworks advancement (CBSD), reusability of a segment is an imperative perspective, which gives the appraisal to reuse the current created part [1]. In the event that a current segment is utilized after fitting appraisal, it lessened the danger, time and expense of the venture improvement process. To reuse the segments, it is important to anticipate or asses the segment reusability with better precision, so that the real playing point of part based frameworks and outline can be taken in a venture. After evaluation of a segment, if part reusability does not turns out to be dependent upon a limit level then it may not respect reuse the segment as it can prompt workaholic behavior and may build the danger, reconciliation time and expense. Because of these sorts of prerequisites in programming improvement process, scientists have been attempting to discover the part reusability utilizing factual and other routine methods. As of late interdisciplinary strategies, for example, fluffy rationale, ANN, Neuro-fluffy have taken lead because of their energy of consistency [2].

II. RELATED WORK ON REUSABILITY

Kumar et al. [3] introduced a depleted survey on quality parts of the segment based frameworks. Reusability is an imperative and principle component of programming quality. The creator led a survey of the examination papers identified with nature of segments construct frameworks situated in light of different variables including practicability, approval evidence and so on and presumed that delicate registering methodologies has not been investigated in this are as such.
In the prior exploration by Kumar et al. [4] gives the opportunity to lessening the product imperfection thickness in a discharge by anticipating it taking into account the information from proposed methodology for the resulting arrivals of a product item. The result demonstrated ANN results better than FIS in anticipating the deformity thickness.

Sharma et al. [5] present the ANN based system to anticipate the reusability of segments. They presumed that more number of segments for may deliver better results for the preparation and testing.

Gill [6] examines the different issues concerning part reusability and its advantages as far as expense and time-reserve funds. Paper likewise gives a few rules to quantify the level of programming reusability in part based improvement. These rules incorporate definite programming reuse evaluation to gauge the potential for rehearsing

reuse, money saving advantage examination to choose whether or not reuse is a beneficial venture, selection of benchmarks for parts to encourage a superior and speedier comprehension of a part and a quicker coordination into a framework, selecting pilot ventures for more extensive advancement of reuse and distinguishing reuse measurements.

Poulin et al. [7] presents an arrangement of measurements utilized by IBM to gauge the endeavors spared by reuse. The study proposes the potential advantages against the uses of time and assets needed to distinguish and incorporate reusable programming into an item. Study accepts the expense as the arrangement of information components like Shipped Source Instructions (SSI), Changed Source Instructions (CSI), and Reused source Instructions (RSI) and so forth. Paper proposes a few other reusability measurements as far as expense and profitability like Reuse expense evasion, Reuse worth included and Additional improvement expense, which can be utilized altogether for business applications.

Cho et al. [8] proposes an arrangement of measurements for measuring different parts of programming segments like intricacy, adaptability and reusability. The work considers two ways to deal with measure the reusability of a segment. The primary is a metric that measures how a segment has reusability and may be utilized at outline stage as a part of a part advancement process. This metric, Component Reusability (CR) is computed by partitioning whole of interface strategies giving shared trait works in an area to the total of aggregate interface techniques. The second approach is a metric called Component Reusability level (CRL) to quantify specific segment's reuse level every application in a part based programming improvement. Notwithstanding, the proposed measurements are in view of lines of codes and must be utilized at configuration time for segments.

Dumke and Schmietendorf [9] proposed an arrangement of reusability measurements for JavaBeans segments. The measurements are adjusted from organized and articles arranged configuration connection and are taking into account the source code of the segments. Thusly, it can't be utilized by part integrators because of the non-availability of the source code.

Washizaki et al [10] talks about the significance of reusability of segments to understand the reuse of parts successfully and propose a Component Reusability Model for discovery segments from the perspective of segment clients. The model recognized elements influencing reusability on the premise of an examination of the exercises did when reusing a blackbox part. The paper additionally directs an observational assessment of these measurements on different Java Bean segments and set certainty interims for these measurements. It additionally creates a relationship among these proposed measurements. These measurements are connected on just for little Java Bean segments and need to be accepted for other segment advancements like .NET, ActiveX and others moreover.

Boxall and Araban [11] considered interfaces of the segments to quantify the reusability. Paper expected that understandability of a segment can be made through its interface properties and understandability influences the level of reuse. Paper additionally proposed a few measurements by considering the extent of the interface, contention check, contention redundancy scale and others. These measurements give a superior comprehension of the properties of segment's interfaces, which may help in measuring the reusability of the part. On the other hand, proposed methodology does not consider alternate viewpoints in the interface, for example, contention unpredictability. A few different scientists likewise viewed as comparative elements for measuring reusability.

Like, Rotaru et al. [12] considered flexibility, form capacity and many-sided quality of a part to depict its reusability.

Mili et al. [13] considered two perspectives, reusability and handiness while REBOOT (Reuse Based on Object-Oriented Techniques) considered elements to be specific convenience, adaptability, understandability and certainty to survey the reusability.

Sagar et al. [14] utilized fluffy rationale to foresee the reusability of part based framework.

Boetticher and Eichmann [15] considered components in particular flexibility, multifaceted nature and coupling and connected neural system based methodology for measuring reusability of discovery Ada parts. Notwithstanding, results got from the experimentation were not exactly calculable with relationship between precise and tested results was just 0.18, which was low.

Acharya and Sadananda [16] proposed Kohnen's self-arranging maps (SOM) based methodology from Artificial Neural Network to sort out different programming parts into groups in light of their attributes to advance programming reuse.

Singh and Saha [17] anticipated the testability utilizing the outline measurements for item arranged programming. Shatnawi and Ziad [18] discussed about numerous oversampling methods that are utilized to enhance the execution of forecast models and proposed to guide the oversampling methodology utilizing the flaw content (i.e., the quantity of flaws in a module).

III. Optimization Techniques Used For Reusability

Numerous techniques for optimization have been in implementation since long for different types of problems. The issue of reusability is an ever challenging problem. The current paper discusses the reusability specifically in the bank loan process in the following sections. Here, two major optimization techniques viz., GA and ACO have been put into use due to their corresponding advantages in terms of accuracy of results.

A. Process example: bank loan process

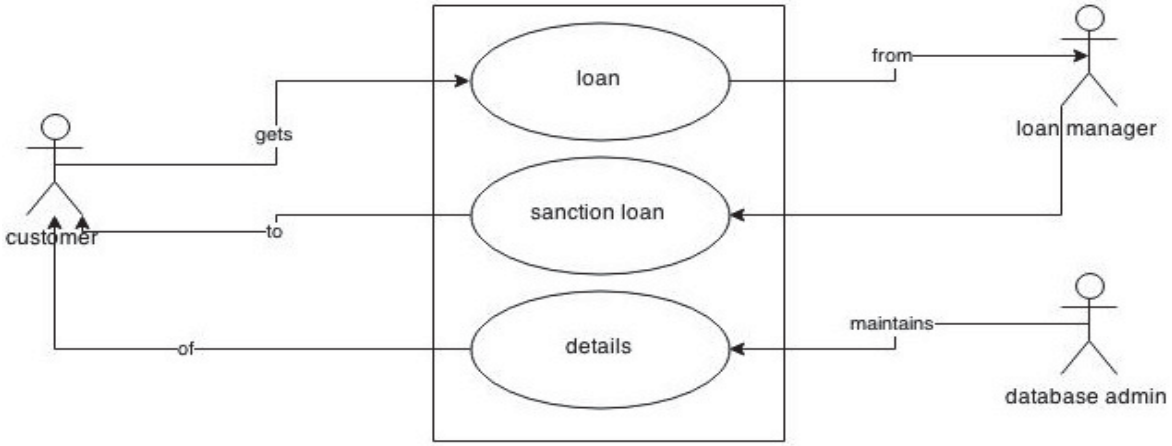

Figure. 1 bank loan process

The use case diagram for credit risk assessment describes the overall activities that can be performed by the employee of the bank. The figure 1 depicts the following steps

1. The employee has to clear the database and should fetch the data
2. Then the customer id is to be matched in the customer data and credit data.
3. With the exposure type the corresponding credit conversion factor is to be matched and multiplied so as to bring the fund based and non-fund based exposures to a normalized same scale.
4. With the asset type and rating the corresponding risk weights for all the advances are found.
5. The amount covered by the security should be detected from the balance.
6. If the security value is greater than the exposure then the risk weighted asset is null else the difference will be the risk weighted asset for the individual advance.
7. The difference value should be detected with the guaranteed value and same like security value should be followed.
8. The risk weighted assets should be classified with asset type and exposure type and the selected report should be segregated and should be generated.

Bank loan process is tied up with several types of risks such as transfer risk, operation risk, liquidity risk, legislation risk, credit risk, operation risk, Market risk and reputation risk. Studies show that main reason for the bank crisis is credit risk [23]. Credit risk defined as when a borrower failed to repay his/her debt. In order to control credit risk and maximize profit commercial banks developed many analytical models to identify potential default loan applicant [22]. Therefore credit risk assessment plays a very important role in identifying the potential default customer. And assessment value can be further reused in the future.

B. Genetic algorithm

Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved. Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according t their fitness - the more suitable they are the more chances they have to reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

Algo_GA

Begin

Generate random population r_p of n chromsomes;

Evaluate fitness $f(x)$ of each chromosome x in the population;

For $x = 1$ to number of generation

Switch (operation)

{

Case crossover:

Select two parents at random x_a and x_b;

x_c = crossover(x_a, x_b); // generate offspring

Break;

Case mutation:

Select two parents at random x_a and x_b;

x_c = mutation(x_a, x_b); // generate offspring

Break;

}

Evaluate the fitness $f(x_c)$;

If ($x_c \geq w_c$) // w_c worst chromosome

{

Replace w_c with x_c

}

Next x;

Check if (termination = true);

End

C. Ant colony optimization algorithm

This algorithm is based on the thoughts of ground dwelling insect scavenging by pheromone correspondence to shape ways. Essentially suited for combinatorial advancement and diagram issues.

Algo_ACO

Begin;

Initialize the (p_t) and (p_{tn}); // pheromone trails and parameters

Generate population p of n solutions (ants);

For each ant $i \in n$:

Calculate fitness $f(i)$;

For each ant determine its best position;

Determine the best global ant;

Update (g_t);

Check if termination = true;

End;

In the current work, the optimization will be done using genetic algorithm first and Ant colony optimization algorithm after that. The genetic algorithm will first selects solution from a number of possible solutions followed by ant colony algorithm which selects from a specific solution set.

IV. Process Defined For Work Organization

A business process or business method is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers. It may often be visualized as a flowchart of a sequence of activities with interleaving decision points or as a Process Matrix of a sequence of activities with relevance rules based on data in the process.

V. Process Defined Languages

A. XML

The XML Process Definition Language (XPDL) is a format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products, i.e. between different modeling tools and management suites. The benefit of XML is that because you are writing your own markup language, you are not restricted to a limited set of tags defined by proprietary vendors.

B. UML

The Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems. UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

C. XAML

Extensible Application Markup Language, or XAML (pronounced "zammel"), is an XML-based markup language developed by Microsoft. XAML is the language behind the visual presentation of an application that you develop in Microsoft Expression Blend, just as HTML is the language behind the visual presentation of a Web page.

VI. Factors Necessary For Reusability

A. Adaptability

Adaptability is the simplicity of change in segment at whatever point required in application. The adaptability is dependent upon desire then the segment might be more reusable furthermore it will be anything but difficult to keep up the part for future rerelease and stages [19].

B. Interface Complexity

Parts can be dealt with as black box, in which we have just ways to get to and use it. At times source code of these parts may be accessible however on the off chance that utilizing the library and connecting, the source code is likewise not accessible.

C. *Understandability*

In the event of source code of the segment is not accessible; the documentation is the best way to comprehend the highlight and interfaces of the part. The documentation helps the client or engineer in segment incorporation and adding to the module to interface with parts.

D. *Versatility*

Versatility is the level of capacity that if there should be an occurrence of environment change, the part has the capacity perform with characterized prerequisite with no or little change at whatever point there is a need from improvement or business viewpoint.

VII. PERFORMANCE PARAMETERS

TABLE I

Parameter	Description
P_{rate}	It is defined as the ratio of correctly detected to the sum of correctly detected plus false positives.
R_{rate}	It is defined as the ratio of the correctly detected to sum of correctly detected plus false negatives.
A_c	It is defined as how close a measured value is to the actual (true) value.
F_m	It is the harmonic mean of the recall and precision rates.

P_{rate} = precision rate

R_{rate} = recall rate

A_c = accuracy

F_m = F-measure

VIII. CONCLUSION

Reusability is a standout amongst the most essential variables for the achievement of any part based programming item. On the off chance that created programming is not exceptionally reusable, then it won't be recommended for the joining in any undertaking in light of the fact that it will build the trouble rather than the effectiveness in programming improvement. We have analyzed in this paper GA, ACO, bank loan process, reusability.

REFERENCES

[1] G. Wang, R. Valerdi, J. Fortune, “Reuse in Systems Engineering”, IEEE Systems Journal, vol. 4, no. 3, pp 376-384, September 2010.

[2] P. S. Sandhu, P. Blecharz, H. Singh, “A Taguchi Approach to Investigate Impact of Factors for Reusability of Software Components”, Journal of World Academy of Science, Engineering And Technology, vol. 25, pp 135-140, Year 2007.

[3] Kumar, V., Sharma, A., Kumar, R. and Grover, P. S., “Quality aspects for component-based systems: A metrics based approach, Software: Practices and Experience”, John Wiley & Sons, December, vol. 42, no. 12, pp: 1531-1548, Year 2012.

[4] Kumar, V., Sharma A. and Kumar. R., “Applying Soft Computing Approaches to Predict Defect Density in Software Product Releases: An Empirical Study”, Computing and Informatics, vol. 32, no.1, pp: 203-224, Year 2013.

[5] Sharma A., Kumar, R., and Grover, P. S., “Reusability assessment for software components”, ACM SIGSOFT Software Engineering Notes, vol. 34, no. 2, March, pp: 1-6, 2009.

[6] Gill, N. S.,” Reusability Issues in Component based Development”, ACM SIGSOFT Software Engineering Notes, vol. 28, no. 6, pp: 30-36, 2013.

[7] Poulin, J., Caruso, J. and Hancock, D., “The Business Case for Software Reuse”, IBM Systems Journal, vol. 32, no. 40, pp: 567-594, 1993.

[8] Cho, E. S., Kim, M. S. And Kim, S. D., “Component Metrics to Measure Component Quality”, 8th Asia-Pacific Software Engineering Conference, Macau, pp: 419-426, 2001.

[9] Dumke, R. and Schmietendorf, A., “Possibilities of the Description and Evaluation of software Components”, Metrics News, Vol. 5, no. 1, pp: 13-26, 2000.

[10] Washizaki, H., Hirokazu, Y., Yoshiaki, F., “A Metrics Suite for Measuring Reusability of Software Components”, Proceedings of the 9th International Symposium on Software Metric, pp: 211-223, 2003.

[11] Boxall M. A. S. and Araban S., “Interface Metrics for Reusability Analysis of Components”, Australian Software Engineering Conference (ASWEC'2004), Melbourne, Australia, pp: 40-46, 2004.

[12] Rotaru, O. P., Dobre, M., Petrescu, M., "Reusability Metrics for Software Components", IEEE International Conference on Computer Systems and Applications (AICCSA-05), Cairo, Egypt, , pp: 24-29, 2005.

[13] Mili, H., Mili, F. and Mili, A., "Reusing Software: Issues and Research Directions", IEEE Transaction on Software Engineering, vol. 21, no. 6, pp: 528-561,1995.

[14] Sagar, S., Nerurkar N. W., Sharma A., "A soft computing based approach to estimate reusability of software components", ACM SIGSOFT Software Engineering Notes, vol. 35 ,no. 5, September pp:1-5,2010.

[15] Boetticher, G. and Eichmann, D., "A Neuro-Fuzzy Based Software Reusability Evaluation System with Optimized Rule Selection", Austrelian Conference on Software Metrics (ACOSM, 93), pp-1-11, 1993.

[16] Acharya, S. and Sadananda, R. , "Promoting Reuse Using Self-Organizing Maps", Neural Processing Letters, no. 5, , pp: 219-226, 1997.

[17] Singh, Y. And Saha, A, "Prediction of testability using the design metrics for object-oriented software", Int. J. of Computer Applications in Technology, vol. 44, no.1, pp.12 – 22.

[18] Shatnawi, R. and Ziad, A., "A guided oversampling technique to improve the prediction of software fault-proneness for imbalanced data", Int. J. of Knowledge Engineering and Data Mining, vol.2, no.2/3, pp.200 – 214, 2012.

[19] C.E.L. Peixoto, J.L.N. Audy, R. Prikladnicki, "Effort Estimation in Global Software Development Projects: Preliminary Results from a Survey," in Proc. 2010 5th IEEE International Conference on Global Software Engineering, ICGSE, pp.123-127, 2010.

[20] M. Jorgensen, B.W. Boehm, "Software Development Effort Estimation: Formal Models or Expert Judgment", IEEE Software, vol. 26, no. 2, pp. 14-19, 2009.

[21] Mili, H., Mili, F. and Mili, A., "Reusing Software: Issues and Research Directions", IEEE Transaction on Software Engineering, vol. 21, no. 6, pp: 528-561, 1995.

[22] Gan, Q., Luo, B., & Lin, C. Z., " Identifying Potential Default Loan Applicants-A Case Study of Consumer Credit Decision for Chinese Commercial Bank 1", SAS global forum, 2008.

[23] Yu, L. Y., Duan, Z. G., & Li, H. L., "A Neural Network Model in Credit Risk Assessment Based on New Risk Measurement Criterion", JCIS, pp. 1102-1105, 2012.