A Framework for Software Safety Analysis with Rough Sets

RICHARD A. WASNIOWSKI
Computer Science Department
California State University Dominguez Hills
Carson, CA 90747, USA

Abstract: - This paper presents a method to evaluate software safety using a rough sets theory. The data about the software product and process are collected via the questionnaire. The result is the direct assessment of the software safety in terms of a single coefficient, whose goodness is analyzed using the rough sets. Java based program has been developed to help in the interactive analysis. Results from a real experiment of software safety evaluation are discussed

Key-Words: - Software Safety, Analysis, Rough Sets

1. Introduction

Today's computer systems involve complexity in both product and process, which results in an increased level of risk in certain classes of applications. If the risk is especially endangering safety of the public, then the computer system is called safety critical. For a safety-critical system, a failure of a computer can cause large-scale catastrophic consequences. This type of systems requires strict consideration of safety features during the development process and in the final product.

The objective of this paper is to investigate the safety aspects of computer software used in safety critical applications. In particular, in this paper, the quantitative evaluation of software safety is attempted. Since safety is a property, which can be hardly quantified, a new theoretical approach is proposed based on the theory of rough sets.

2. Problem description

Software safety is an important issue in safety critical systems. The idea behind assessing and improving software safety is to ensure that the software controlled system will execute without resulting in an inadmissible risk [6]. A defect in software can lead to a failure. The consequences of a failure in a software- controlled critical application may be extremely severe, often creating threat to a human life.

Various attempts to evaluate safety on a quantitative scale have not been very successful. Rather than that, measures to assess safety use qualitative evaluations based on verbal, subjective judgments. Such data, even carefully taken to minimize subjectivity during the collection process, are hard, near impossible, to analyze because they lack probabilistic characteristics
and, thus, the analysis cannot be based on statistical methods.

3. Methodology

The problem of evaluating safety of software is split into three steps: (1) Selecting the method to collect data on soft ware quality, (2) Developing a theoretical method for data assessment and (3) Conducting a real experiment.

In this study, the questionnaire was designed based on the principles discussed in [13]. This questionnaire applies to the soft-ware development process. It is divided into six parts based on the software development

life-cycle: project planning, specification of requirements, design, implementation and integration, verification and validation and operation and maintenance.

In addition to that, a list of the most important questions called the screen questions has been isolated to provide a preliminary evaluation, in case it is clear that a detailed analysis is not needed. Answers to these screen questions must be satisfied first, before proceeding with more detailed questions, Answering both the screen and detailed questions is required to reason about the safety.

Rough sets theory [7] is a mathematical technique developed to describe quantitatively uncertainty, imprecision, and vagueness. It seems therefore suitable to describe the concept of safety, because safety is a vague concept in itself. Moreover, to answer the question whether a system is safe or not a sharp answer, yes or no, can hardly be given. Rough sets are used in this work for the analysis of the imprecise data collected from the questionnaire to determine how safe is the software developed, Below a brief outline is provided of the rough sets theory and the way we deal with uncertainty concepts to make quantitative decisions.

4. Theoretical background

The rough set theory, introduced by Z. Pawlak provides sound mathematical tools to deal with inconsistencies in data sets. The rough set approach does not require any preliminary or additional information about data such as probability in probability theory or grade of membership in fuzzy set theory. In the rough set theory inconsistencies are not removed from consideration. Instead, lower and upper approximations of the concept are computed. An information system (IS) consists of a set of objects. IS is usually represented as a pair, S = (U, A), where U and A are finite, nonempty sets called the universe and a set of attributes. The domain of a, denoted Va, is associated with every attribute, a∈ A. Then, an indiscernibility relation is defined as a binary relation I(B) on U for any subset B of A: (x, y) ∈ I(B) if and only if a(x) = a(y) for every a ∈ A, where a(x) denotes the value of attribute a for element x. I(B) is an equivalence relation on U, the partition determined by B is denoted by U/I(B) or simple U/B. An equivalence class of I(B) containing x is denoted by B(x), and objects x and y are B-indiscernible if (x, y) belongs to I(B). Equivalence class of the relation I(B) are referred to as B-elementary sets or B-granules.

Approximations are operations assigning every X⊆ U to two sets BX and $\overline{B}X$ called the B-lower and the B-upper approximation of X, respectively:

$$\underline{B}X = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\},$$

$$\overline{B}X = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \phi\},$$

The B-lower approximation of X is a collection of objects that can be classified certainly to the set of X using the attributes set B. On the other hand, B-upper approximation of X is the collection of objects that can possibly be classified to the set X.

A rough set is defined using the boundary region between those approximations:

$$BN_B(x) = \overline{B}X - \underline{B}X,$$

which is referred to as the B-boundary of X. If the boundary region of X is an empty set (BNB(x) = ϕ), then X is crisp (exact) with respect to B. Otherwise, X is referred to as rough (inexact) with respect to B.

Degree of roughness is characterized numerically with the following:

$$\alpha_B(X) = \frac{card(\underline{B}X)}{card(\overline{B}X)}$$

$0 \leq \alpha B(X) \leq 1$

If αB(X) < 1, X is rough with respect to B. Otherwise, X is crisp.

Another important measure is defined using a rough membership function:

$$\mu_B(x) = \frac{card(B(x) \cap X)}{card(B(x))}$$

$0 \le \mu B(x) \le 1$

The rough membership function shows how strongly an element x belong to the rough set X with the set of attributes B. In other words, the rough membership function expresses a degree of certainty to which x belongs to X. A decision table, a special form of the information system, is represented with $T = (U, A, C, D)$, where C and D are subsets of A called condition and decision attributes, respectively. The subsets show the following properties: $C \cap D = A$ and $C \cup D = \phi$.

If all values of attributes from D are uniquely determined by values of attributes from C, a decision table is consistent, which is denoted $C \Rightarrow D$. Otherwise, the decision table is inconsistent. In a consistent decision table, a set of attributes D depends totally on a set of attributes C, and there exists a functional dependency between values of C and D. Otherwise, a set of attributes D depends partially on a set of attributes C. Dependency measure is a basic issue in data mining in that it reveals relationships in a database and is related with approximations. If C and D are subsets of A, then dependency is defined with the following equation:

$$k = \gamma(C, D) = \sum_{X \in U/D} \frac{card(\underline{C}X)}{card(U)}.$$

If k = 1, D depends totally on C. Otherwise, D depends on C to a degree k, which is denoted $C \Rightarrow_k D$. The coefficient k is the ratio of all elements of the universe, which can be properly classified to blocks of the partition U/D, using attributes C. A reduct is "a minimal set of condition attributes that preserves the degree of dependency. Some redundant attributes can be removed from the IS without loss of information based on the concept of a reduct. However, some attributes (generally the attributes in CORE) cannot be removed to keep the information, which means an attribute has different degree of significance from another one.

5. Applying rough sets theory

In this study, the questionnaire was designed based on the principles discussed in [13]. This questionnaire applies to the software development process. It is divided into six parts based on the software development life-cycle: project planning, specification of requirements, design, implementation and integration, verification and validation and operation and maintenance.

The questions are related to the safety aspects of development activities and techniques in each phase. However, the questionnaire includes also other areas, such as risk management, reliability, security, etc.

Rough sets theory [7] is a mathematical technique developed to describe quantitatively uncertainty, imprecision, and vagueness. Once the data have been collected, the so called screen evaluation has to be done [6]. This term describes the basic group of activities that must exist to preserve safety in the product and process. It involves a series of general questions that are crucial within the software project. These questions need to be answered and pass the test in order to continue execution of the software safety assessment. The meaning of this step is to make the determination whether to continue or abandon the evaluation if the screen evaluation level exceeds or not a certain threshold value.

In the second step, the detailed evaluation is conducted, based on answers to detailed questions. An answer to each detailed question is transformed into a numerical value from the range [0-1] and the weighed sum of all answers is calculated, with weights, weight(i), representing the criticality of a respective question:

answer(i) x weight(i).

This number representing safety is meaningless if not evaluated for trustworthiness or accuracy. Here the rough

sets theory comes into play, allowing associating some kind of confidence with this single number. Basically, we are seeking an estimate, how accurate is the safety evaluation.

For a particular phase of the software development cycle a set of equivalence relations is created based on the similarity of values S, called decision attributes. Then, in the universe of all cases (different evaluations), the rough set is created spanning over a range of values S. Using the rough sets techniques [5], not described here due to the lack of space, the aR(X) coefficient is calculated to determine how accurate is the approximation. The details of the analysis are presented in the report [15].

6. Experiments

The results of this study show that using rough sets theory for safety evaluation of software is a valuable means of improving overall quality of the product and process, as well as assessing trustworthiness of the results of safety evaluation. The safety property can be measured in terms of the attributes represented by a questionnaire. The approach is flexible, so that, in principle, various checklists can be used. Questions are as general as possible, so the assessment can be applied to different applications. In this respect, the method does not depend on specific applications.

Mathematical evaluation of the results, supported by an automatic tool, can lead to the detection of weak points in the development cycle and help in improving the organization of the software project. The outcome of the safety evaluation shows the areas or activities of a safety-critical software project that might need work to provide a safer system. The rough sets analysis determines the overall safety level of the project and shows how good the approximation of the safety evaluation is. This provides a way to find out whether the overall safety of the project is sufficient or there is need to improve the process.

It should be realized that the main purpose of this approach is to help establish whether the software is safe, not to determine safety beyond doubt. The procedure to evaluate software with respect to safety should be taken as a part of the assessment process, not as the final product of the assessment phase. To fully assess the capability of the method, it has to be tested on more realistic data from real projects.

7. Conclusion

The results of this study show that using rough sets theory for safety evaluation of software is a valuable means of improving overall quality of the product and process, as well as assessing trustworthiness of the results of safety evaluation. The safety property can be measured in terms of the attributes represented by a questionnaire. The approach is flexible, so that, in principle, various checklists can be used. Questions are as general as possible, so the assessment can be applied to different applications. In this respect, the method does not depend on specific applications.

8. References

[1] Bishop, C. M., Neural Networks for Recognition:Oxford University Press, 1995.

[2] Chen-Jimenez I.E., Software Assessment Tool for Evaluation and Analysis of Safety-Critical Applications Using Rough Sets Theory, Master of Software Engineering Project, Embry-Riddle Aeronautical University, Daytona Beach, Fla., 1997

[3] Clayton C., A Checklist for Software Safety, Society for Software Quality, San Diego, Calif., 1996,

[4] Grzymala-Busse, J. “A new version of the rule induction system LERS”, Fundamenta Informaticae, Vol. 31, No. 1, 1997, 27-39

[5] Lawrence J.D. (Ed.), Proc. Workshop on Developing Safe Software, San Diego, Calif.,

July 2223, 1992, Report NUREG/CP-0145, U.S. Nuclear Regulatory Commission, Washington, DC, November 1994

[6] Leveson N., Safeware: System Safety and Computers, Addison-Wesley, Reading, Mass., 1995

[7] Pawlak Z., Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, 1991

[8] Pawlak, Z., "Rough sets", International Journal of Information & Computer Sciences 1, 341-356, 1982

[9] Pawlak, Z., and Slowinski, R., "Rough set approach to multi-attribute decision analysis", European Journal of Operational Research 72 443 – 459,1994

[10] Ohrn, A., Komorowski, J., Skowron, A., Synak, P., “A software system for rough data analysis”, Bulletin of the International Rough Sets Society, Vol. 1 , No. 2, 1997, 58-59

[11] Ras, Z.W., Joshi, S., Ras, Z., Joshi, S., “Query approximate answering system for an incomplete DKBS", in Fundamenta Informaticae Journal, IOS
Press, Vol. 30, No. 3/4, 1997, 313-324

[12] Persons W.L., The Safety-Critical Software Evaluation Assistant (SEA), Proc. 2nd IFAC Workshop Safety and Reliability in Emerging Control Technologies, Daytona Beach, Fla., November 1-3, 1995, pp. 95-100, T. Hilburn et al. (Eds.), Pergarnon, Oxford, 1996

[13] Redmill F. (Ed.), Dependability of CriticalComputer Systems, Vols 1-2, Elsevier, London, 1988/89

[14] Wasniowski R., “Improving the Performance of Patterns Recognition with Neural Networks,” RAW-94- 32, June, 1994.

[15] Wasniowski R., , “The Use of Rough Sets in Data Mining”, RAW-96-14, June, 1996.