Subjective Safety Analysis for Software Development

J. Wang[1], A. Saeed[2] and R. de Lemos[2]

[1]Department of Engineering and Technology Management
Liverpool John Moores University L3 3AF, UK
[2]Centre for Software Reliability, Department of Computing Science
University of Newcastle upon Tyne NE1 7RU, UK

ABSTRACT

This paper presents a framework for subjective safety analysis of software requirements specifications for safety−critical systems. The framework incorporates fuzzy set modelling and evidential reasoning to assess the safety associated with safety requirements specifications. Fuzzy set theory is used to model each safety rule and evidential reasoning is employed to synthesize the information produced. Three basic parameters − failure likelihood, consequence severity and failure consequence probability are used to analyse a safety rule (a basic element of a software requirements specification) in terms of membership functions. The subjective safety description associated with the safety rule is then mapped back to a scale of pre−defined safety expressions which are also characterised in terms of membership functions. Such a mapping results in the production of the safety evaluation associated with the safety rule, expressed in terms of the degrees to which the subjective safety description belongs to the pre−defined safety expressions. Such degrees represent uncertainty in the safety evaluation associated with the safety rule. The information produced for all safety rules can then be synthesized using an evidential reasoning approach to obtain the safety evaluation associated with the safety requirements specifications. The developed framework is capable of dealing with multiple safety analysts who make judgements on each safety rule.

KEYWORDS

Fuzzy sets, software safety analysis, subjective safety analysis, evidential reasoning, information models and formal notations.

1. INTRODUCTION

The increased employment of computer−based systems for the implementation of critical functions has introduced new challenges for the development and *assessment* of software. For assessment, evidence must be provided to demonstrate that the risk associated with the software is acceptable within the overall system risk, IEC (1992). It has been proposed that an effective approach to assess and reduce the contribution of software failures to system risk is to conduct safety analysis in parallel with the phase of requirements analysis, within the software development lifecycle, Saeed *et al* (1995). In accordance with the proposed approach, the outputs of the requirements analysis are safety requirements specifications for the software, expressed in a formal notation. An information model is used as a structure to record the relationships between critical failure behaviours of the overall system (i.e. accidents and hazards) and the safety requirements specifications (safety constraints and safety strategies) for the software, de Lemos *et al* (1995). The results of the safety analysis provide arguments which support the validity of the relationships encoded in an instance of the information model, thereby evidence that the risk posed by software is acceptable.

Safety analysis can be conducted on a qualitative or quantitative bases. Qualitative safety analysis aims to confirm that under normal circumstances the safety requirements specifications will prevent the system to enter into a hazard state, and examine the impact on hazards of defects in the specifications and violations of associated assumptions. Qualitative safety analysis can be conducted effectively by applying formal verification techniques and safety analysis techniques, Saeed *et al* (1994). Quantitative safety analysis aims to deal with the limitations of qualitative safety analysis, by providing a measure of the safety associated with the safety requirements specifications. For software development the measures should identify if the risk associated with a specification is acceptable, and when alternative specifications are proposed provide a basis for decision making. For traditional technologies, quantitative analysis is conducted in terms of probability distributions of primitive failure events. However, it is difficult to determine precisely probability distributions for those issues which can affect software safety. A novel approach pursued in this work is to express uncertainty in the safety associated with safety requirements specifications in terms of vague and imprecise descriptors like '*reasonably low*', terms commonly used by safety analysts that can be expressed in fuzzy set theory, Wang *et al* (1995).

This paper proposes a framework for subjective safety analysis of requirements specifications, based on fuzzy set modelling and evidential reasoning, Wang *et al* (1995). Deductive analysis starts with the stipulation of acceptable levels of safety for each accident as a *linguistic variable* (a pre−defined fuzzy safety expression) and dictates acceptable levels of safety for the hazards, from which a stipulated risk level (a numerical measure) is calculated. An alternative is to determine a linguistic variable for a hazard, on the basis of a traditional estimate of acceptable risk for that hazard. The risk of a hazard is controlled by the safety strategies defined to maintain the safety constraint that will exclude the hazard. The safety strategies are defined in terms of *safety rules*, which are based upon *assumptions* (also expressed formally), under which safe behaviour is maintained; these rules are characterized as primitive elements of a safety strategy. Inductive analysis starts with the application of fuzzy set theory to analyse these elements using three basic parameters, *failure likelihood*, *consequence severity* and *failure consequence probability*, in terms of membership functions. The subjective safety description associated with an element is mapped back to a scale of pre−defined safety expressions, to determine the uncertainty safety evaluation (i.e. the extent to which the rules belongs to each expression on the scale) for each element. The safety evaluations for each element are synthesized using evidential reasoning to obtain the safety evaluation for a safety strategy. These safety expressions can be used to rank alternative safety strategies, supporting development decisions for risk reduction. A similar synthesis process is used to obtain the safety expression of a safety constraint from the safety expressions of associated safety strategies. An estimated risk level is then computed for each safety constraint and compared with the stipulated risk level for the associated hazard, to confirm that the risk is acceptable. As the development proceeds, the safety strategies will be refined into more detailed specifications and the additional information can be used to re−evaluate the initial assessments and further direct risk reduction. The approach is capable of dealing with evidence from diverse sources, such as multiple safety analysts who make judgements on safety based on the results of different techniques. The feasibility of the framework was illustrated by application to a railway safety problem, Wang *et al* (1996).

2. THE ANALYSIS OF SOFTWARE SAFETY REQUIREMENTS

The framework for subjective safety analysis, presented in this paper, is described in the context of a systematic approach to the analysis of safety requirements, Saeed *et al* (1995). The systematic approach partitions the analysis into smaller phases; each phase corresponds to a domain of analysis (a particular scope of the analysis, e.g. a component of the system) in which requirements analysis and safety analysis are conducted in parallel. The results of applying the approach are encoded in an information model, the Safety Specification Graph (SSG), which records the safety requirements specifications obtained in each phase, such as accidents, (AC) hazards (HZ), safety constraints (SC − a condition that negates a hazard) and safety strategies (SS − a scheme to maintain a safety constraint) and their logical relationships. An SSG is represented as a linear graph, in which a *node* represents a safety specification and an *edge* denotes that a relationship exists between a pair of safety specifications. For a system for which *I* accidents have been identified, the SSG consists of *I* component graphs, one for each accident. The SSG records three kinds of relationships:

- *Coverage*. Absence of all hazards associated with an accident ensures that the accident does not occur.
- *Exclusion*. A safety constraint excludes all the associated hazards.
- *Refinement*. A safety strategy maintains all the specifications of the previous layer to which it is linked.

Two characteristics of an SSG that make it amenable to subjective safety analysis are that the requirements specifications are expressed in a formal notation and the logical relationships between them are explicitly encoded. This supports a better judgement over factors related to a single specification and factors dependent upon the interrelationships between specifications, respectively.

3. A FRAMEWORK FOR SUBJECTIVE SAFETY ANALYSIS OF SOFTWARE SAFETY

A framework for hierarchical subjective safety analysis of safety requirements specifications, for the initial layers of an SSG is proposed as shown in Figure 1; an *ellipse* represents the safety evaluation of the named specification and an *arrow* gives the propagation direction of safety information from one level to another. The safety analysis is comprised of a top down process and a bottom up process. The top down process leads to a stipulated risk level for each hazard. This expression is then used to derive acceptable levels of safety for the hazards associated to the accidents via a *coverage* relationships. The bottom−up process starts with associating safety evaluations with the safety rules at level 5, these are then used to determine the safety evaluations associated with the safety strategies at level 4 which further determine the safety evaluations associated with the corresponding safety constraints at level 3. Between levels 2 and 3 a comparison is conducted between the safety values stipulated with hazards and those safety constraints that aim to *exclude* the hazards, this is used to determine if the safety associated with the requirements specifications is acceptable. The framework consists of three main activities: the *stipulation* of a safety level for each hazard, the *estimation* of safety evaluation for each safety constraint and a *comparison* between the stipulated and estimated description of safety. In this paper, we will focus on the approach to the estimation of safety (see section 4).

3.1 Stipulation of Safety

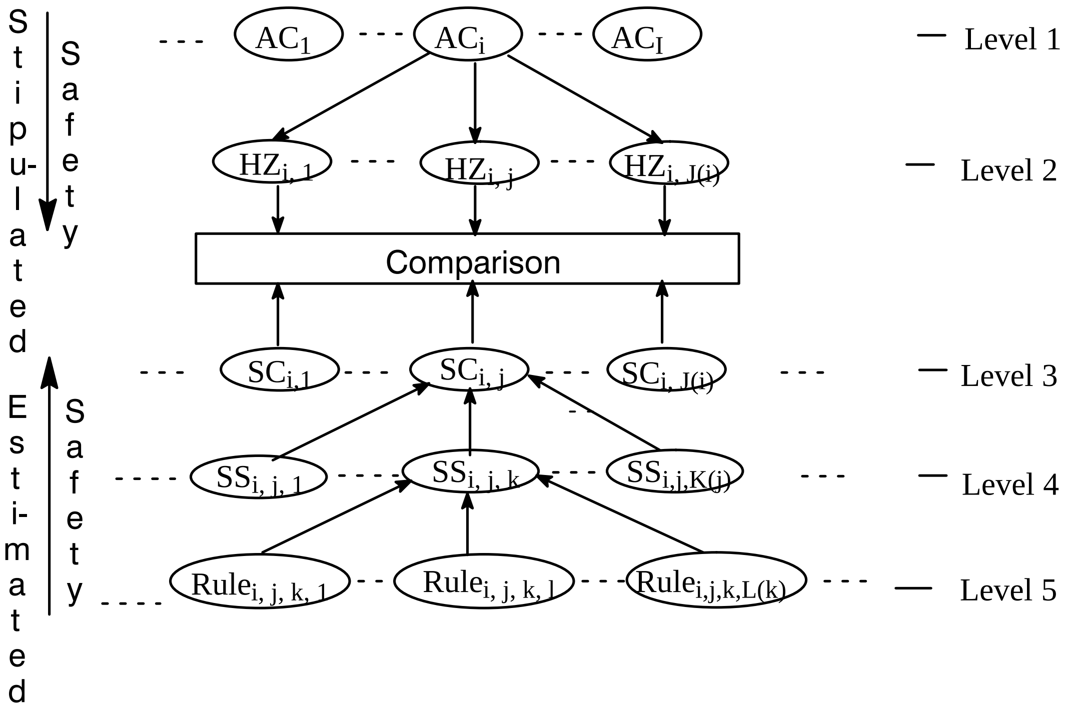
The safety associated with the safety requirements specifications should be contained to a level that is acceptable, depending on the particular situation in hand. This requires that the risk level associated with $HZ_{i,j}$ be stipulated, it commonly understood that safety can be described using linguistic variables, such as '*poor*', '*fair*', '*average*' and '*good'* that are referred to as safety expressions (see section 4.1.4) and provide a scale. The procedure used to associate a safety expression with a hazard will depend on the situation, it can be derived from a safety expression of the accident AC_i or from a traditional estimate of acceptable risk for the hazard. To obtain the level of risk associated with $HZ_{i,j}$ in terms of numerical values for comparison purposes, it is necessary to described the linguistic variable using numerical values. The numerical values associated with the four defined safety expressions can be calculated by studying the categories and membership values associated with the safety expressions.

3.2 Estimation of Safety

Fuzzy set modelling is used to produce the safety evaluation associated with each safety rule at the bottom level, and evidential reasoning is employed to implement the hierarchical evaluation at different levels. The application of evidential reasoning avoids any information loss which may occur in the hierarchical evaluation of fuzzy information using fuzzy set theory, Anderson (1988) and Keller and Kara−Zaitri (1989).

3.3 Comparison of Safety

The comparison of the stipulated risk level and the estimated risk level associated with $SC_{i,j}$ can then be carried out to see if the risk is acceptable, by converting the estimated safety



Key: AC_i – accident i, I number of accidents,
$HZ_{i,j}$ – hazard j of AC_i , $J(i)$ number of hazards of AC_i,
$SC_{i,j}$ – safety constraint j of $HZ_{i,j}$,
$SS_{i,j,k}$ – safety strategy k of $SC_{i,j}$, $K(j)$ number of strategies for $SC_{i,j}$,
$Rule_{i,j,k,l}$ – safety rule l associated with $SS_{i,j,k}$,, and $L(k)$ number of rules for $SS_{i,j,k}$.

Figure 1: A hierarchical framework for subjective safety analysis.

evaluation to a numerical value using the same scale as used for determining the stipulated level. If the risk associated with $SC_{i,j}$ is acceptable, the produced information can be used as evidence to support certification, otherwise it may be required to modify the safety requirements specifications to increase the level of safety. After modifications some parts of the safety analysis will need to be conducted again to make sure that the required level of risk has been contained.

4. APPROACHES FOR SUBJECTIVE ESTIMATION OF SAFETY

To provide a subjective estimate for the safety of software requirements specifications, fuzzy set modelling techniques are used to model the judgements of safety analysts and evidential reasoning is used for the hierarchical propagation of the safety judgements. The main activities of the overall process are illustrated by the SADT diagram in Figure 2.

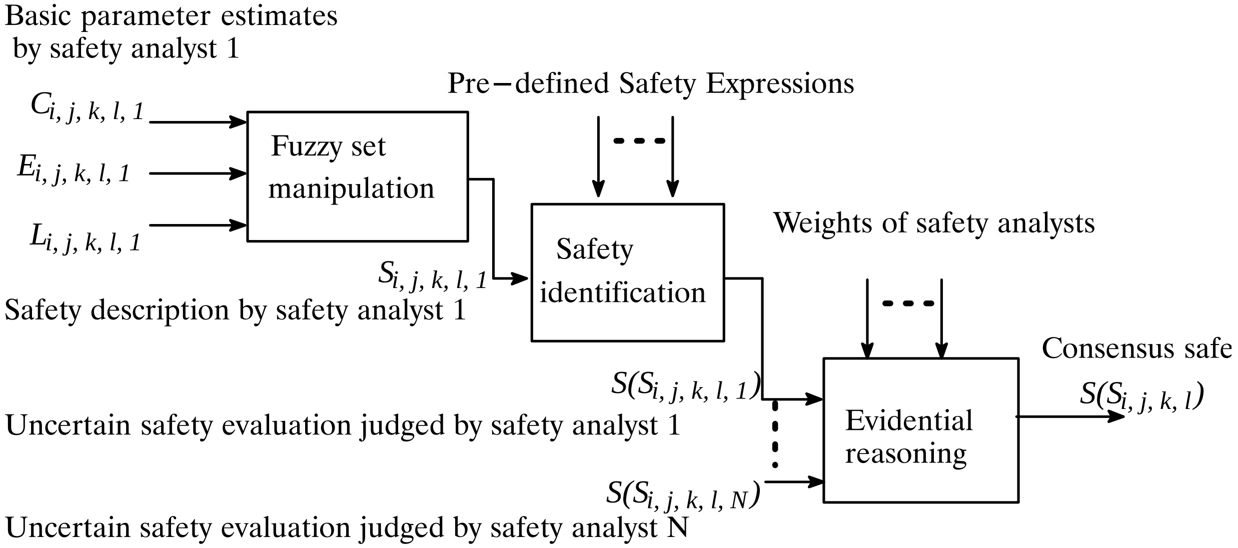

Basic parameter estimates by safety analyst 1
$C_{i, j, k, l, 1}$
$E_{i, j, k, l, 1}$
$L_{i, j, k, l, 1}$
Fuzzy set manipulation
$S_{i, j, k, l, 1}$
Safety description by safety analyst 1
Pre−defined Safety Expressions
Safety identification
$S(S_{i, j, k, l, 1})$
Uncertain safety evaluation judged by safety analyst 1
$S(S_{i, j, k, l, N})$
Uncertain safety evaluation judged by safety analyst N
Weights of safety analysts
Evidential reasoning
Consensus safe
$S(S_{i, j, k, l})$

Figure 2: Activities for the Subjective Estimation of Safety.

4.1. Fuzzy Set Modelling: Safety Definition

The safety associated with a safety rule (say, $Rule_{i, j, k, l}$) can be modelled by studying the associated failure likelihood, consequence severity and failure consequence probability as described earlier. These three parameters can be described by linguistic variables which can be further described by membership functions. A membership function is a description which consists of membership values to categories. The typical linguistic variables for describing *failure likelihood*, *consequence severity* and *failure consequence probability* may be defined in terms of membership degrees belonging to the seven categories, as recommended in Karwowski and Mital (1986), for details of our definitions see Wang *et al* (1996). The membership degrees of the typical linguistic variables are not exclusive with respect to a category, this makes it easier for safety analysts to make judgements on a safety rule. It is obviously possible to have some flexibility in the definition of membership functions for the typical linguistic variables to suit different situations.

4.1.1 Local Safety Parameter

The failure likelihood can be assigned by a safety analysts examining a safety rule, specifically by estimating the likelihood that the safety rule will be violated. To estimate the failure likelihood, for example, an analyst would use such variables as *'highly frequent', 'frequent', 'reasonably frequent', 'average', 'reasonably low', 'low'* and *'very low'*.

4.1.2 Global Safety Parameters

The consequence severity and the failure consequence probability are parameters derived from specifications at higher layers. To estimate the consequence severity, an analyst would use such variables as '*catastrophic', 'critical', 'marginal*' and '*negligible*'. The consequence severity can be assigned studying the severity class of the potential accident caused by the violation of a safety rule (in fact, it should be the same for all safety rules connected to an accident). However, it may be comparatively difficult for safety analysts to assign membership degrees for the failure consequence probability, described using variables, such as '*definite', 'highly likely', 'reasonably likely', 'likely', 'reasonably unlikely', 'unlikely'* and '*highly* unlikely'. This is because it may be required to study the logical relations between safety strategies and between hazards leading to the accident.

The failure consequence probability for safety rule $Rule_{i, j, k, l}$ is denoted by $E_{i, j, k,l.}$ Four conditional properties need to be estimated to determine $E_{i, j, k,l}$, $e_{i, j, k, l} - PSS_{i, j, k}$ is violated if $Rule_{i, j, k, l}$ is violated, $e_{i, j, k} - SC_{i, j}$ is violated given that $PSS_{i, j, k}$ is violated, $e_{i, j} - HZ_{i, j}$ occurs if $SC_{i, j}$ is violated, and $e^{HZ}_{i, j}$ AC_i happens if $HZ_{i, j}$ occurs. Multiple analysts may be involved in the identification of the individual conditional probabilities. The failure consequence probability is estimated on the basis of these probabilities; for example, if $e_{i, j, k, l}$, $e_{i, j, k}$, $e_{i, j}$ and $e^{HZ}_{i, j}$ are all estimated as '*low*', then the literal estimate for $E_{i, j, k,l}$ would be '*low*'. Obviously experience, together with an appreciation of the logical structure of the SSG, would enable a more informed assignment of membership degrees of the failure consequence probability.

4.1.3 Combination of Parameters

Suppose $L_{i, j, k, l}$ represents the fuzzy set of the failure likelihood of occurrence associated with $Rule_{i, j, k, l}$ (i.e the likelihood that $Rule_{i, j, k, l}$ is violated) and $C_{i, j, k, l}$ represents the fuzzy set of the consequence severity. The subjective safety description $S_{i, j, k, l}$ for $Rule_{i, j, k, l}$ can be defined as in (1), Karwowski and Mital (1986), where symbol 'o' represents the composition operation and '×' the Cartesian product operation.

$$S_{i, j, k, l} = C_{i, j, k, l} \text{ o } E_{i, j, k, l} \times L_{i, j, k, l} \tag{1}$$

The relationship between the membership functions associated with $S_{i, j, k, l}$, $C_{i, j, k, l}$, $E_{i, j, k, l}$ *and* $L_{i, j, k, l}$ is:

$$\mu_{S_{i,\ j,\ k,\ l}} = \mu_{C_{i,\ j,\ k\ l}} \text{ o } \mu_{E_{i,\ j,k,l}} \times \mu_{L_{i,\ j,\ k,\ l}} \tag{2}$$

where $\mu_{S_{i,\ j,\ k,\ l}}$ is the membership function for $S_{i,\ j,\ k,\ l,}$ and the others terms are similarly defined.

4.1.4 Fuzzy Safety Identification

To evaluate $S_{i, j, k, l}$ in terms of the basic safety expressions, it is necessary to characterize them using membership degrees with respect to the same categories used in order to map the obtained subjective safety description back to the pre−defined safety expressions. When characterizing the safety expressions, the conditions such as (3) need to be satisfied to confine the safety expression space within the certain extent, for details see Wang *et al* 1996.

$$\mu_{S^{poor}_{i,\ j,\ k,\ l}} = \mu_{C^{catastrophic}_{i,\ j,\ k\ l}} \text{ o } \mu_{E^{definite}_{i,\ j,k,l}} \times \mu_{L^{frequent}_{i,\ j,\ k,\ l}} \tag{3}$$

The variables *'poor', 'fair', 'average'* and *'good'* are described by safety expressions *(m= 1, 2, 3* or *4)*, respectively. Each fuzzy expression is defined as a set of seven pairs,the first elements is the membership category and the second the membership degree.

$$1.\, poor \;=\{(1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0.75), (7, 1)\} \tag{4}$$

$$2.\, fair \;=\{(1, 0), (2, 0), (3, 0), (4, 0.5), (5, 1), (6, 0.25), (7, 0)\} \tag{5}$$

$$3.\, average \;=\{(1, 0), (2, 0.25), (3, 1), (4, 0.5), (5, 0), (6, 0), (7, 1)\} \tag{6}$$

$$4.\, good \;=\{(1, 1), (2, 0.75), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0)\} \tag{7}$$

The extent to which $S_{i,\, j,\, k,\, l}$ belongs to the mth (m = *1, 2, 3* or *4*) safety expression can be obtained using the Best–Fit method, Schmucker (1984), described by $\beta^{m}_{i,\ j,\ k,\ l}$ (m = *1, 2, 3* or *4*).

4.2. ***Evidential Reasoning: Hierarchical Propagation for Safety Synthesis***

Evidential reasoning is used to synthesize the judgements of different safety analysis, in order to determine a safety evaluation for each rule, and then to propagate the safety evaluations up the levels of the framework.

4.2.1 Fuzzy Set Modelling by Multiple Safety Analysts

If multiple safety analysts are involved in the safety analysis process, their judgements need to be synthesized. A diagram for synthesizing the judgements on a safety rule produced by multiple safety analysts is shown in Figure 2. Suppose there are N safety analysts who assign membership degrees for three basic safety parameters associated with a safety rule. Suppose $L_{i, j, k, l, n}$, $C_{i, j, k, l, n}$ and $E_{i, j, k, l, n}$ represent the three basic safety parameters associated with $Rule_{i, j, k, l}$ judged by safety analyst n ($n = 1, \cdots,$ or N), respectively. The subjective safety description $S_{i, j, k, l, n}$ associated with $Rule_{i, j, k, l}$ judged by safety analyst n can be obtained:

$$S_{i, j, k, l, n} = C_{i, j, k, l, n} \circ E_{i, j, k, l, n} \times L_{i, j, k, l, n} \tag{8}$$

$S_{i, j, k, l, n}$ ($n = 1, \cdots,$ or N) can be mapped back to the defined safety expressions to identify the uncertainty safety evaluation $S(S_{i, j, k, l, n})$ associated with $Rule_{i, j, k, l}$, as judged by safety analyst n. Suppose $\beta^{m}_{i,\ j, k, l, n}$ (m = *1, 2, 3* or *4*) represents the extent to which $S_{i, j, k, l, n}$ belongs to the mth safety expression. $S(S_{i, j, k, l, n})$ can be expressed in the following form:

$$S(S_{i, j, k, l, n}) = \{(\beta^{1}_{i, j, k, l, n}, \text{'poor'}),\ \ (\beta^{2}_{i, j, k, l, n}, \text{'fair'}),\ \ (\beta^{3}_{i, j, k, l, n}, \text{'average'}),\ \ (\beta^{4}_{i, j, k, l, n}, \text{'good'})\} \tag{9}$$

It is required to synthesize all $S(S_{i,\ j,\ k,\ l, n})$ (n = *1, ...,and N*) to obtain the safety evaluation associated with $Rule_{i, j, k, l}$. An evidential reasoning approach can be employed to synthesize $S(S_{i,\ j,\ k,\ l, n})$ (n = *1, ... , and N*) and take into account the weight of each safety analyst without losing any useful safety information.

Evidential reasoning is well suited for handling uncertain and inconsistent safety evaluations, Yang and Sen 1994, and is based on the principle that it will become more likely that a given hypothesis is true if more pieces of evidence support that hypothesis. In Figure 2, whether the safety evaluation associated with a safety rule belongs to '*poor*', '*fair*', '*average*' or '*good'* can be regarded as a hypothesis. If the judgement on a safety rule produced by a safety analyst is to some extent evaluated as '*good'*, for example, then the safety associated with the safety rule would be to some extent evaluated as '*good',* depending on the judgement itself and the weight of the safety analyst in the evaluation process. The application of the evidential reasoning approach provides a systematic way of synthesizing such uncertain safety evaluations involving multiple analysts' judgements to produce the safety evaluation for a safety rule.

4.2.2 Hierarchical Propagation of Safety Evaluations

After the safety evaluation associated with each safety rule has been obtained, the safety evaluations associated with all rules − $Rule_{i,\,j,\,k,\,l}$ $(l = 1, \cdots, R(K))$ are synthesized to obtain the safety evaluation associated with $SS_{i,\ j,\ k}$. Then the safety evaluations produced for all $SS_{i,\ j,\ k}$ $(k = 1, \cdots,$ and $K(j))$ are synthesized to obtain the safety evaluation associated with $SC_{i,\,j}$.

5. CONCLUSIONS

A framework incorporating fuzzy set modelling and evidential reasoning is proposed for subjective safety analysis of software requirements specifications for safety−critical systems. In this framework, a fuzzy set modelling method is used to analyse the safety associated with a safety rule, which is judged in terms of three basic parameters by multiple safety analysts. An evidential reasoning approach is then used to synthesize the information produced to obtain the safety evaluation associated with the safety requirements specifications. Finally, a comparison is made between the estimated safety evaluation and the stipulated risk level. The proposed framework can be used as an alternative approach for analysts to conduct safety analysis for software specifications, especially in the situations where there is a lack of quantitative safety data for use in probabilistic risk analysis and where non−numerical safety data is dealt with. Enhancements to the approach include conducting a Failure Mode, Effects and Criticality Analysis (FMECA) of each safety rule and then employ fuzzy set modelling at the failure mode level. This may make it more effective and efficient for safety analysts to make judgements. Other factors such as assumptions on the basis of which specifications are produced may also need to be taken into account to increase the effectiveness of the framework in order to facilitate more practical applications.

ACKNOWLEDGEMENT

The authors acknowledge the financial support of COPERNICUS Joint Research Project ISAT, and thank Dr. J. B. Yang of Birmingham University for allowing use of his software.

REFERENCES

Andersson, L. (1988). *The Theory of Possibility and Fuzzy Sets: New Ideas for Risk Analysis and Decision Making*, Swedish Council for Building Research.

de Lemos, R., Saeed A. and Anderson, T. (1994). On the safety analysis of requirements specifications, Proceedings of *13th International Conference on Computer Safety, Reliability and Security (SAFECOMP'94)*, Ed. Victor Maggioli, Anaheim, CA, 217–227.

de Lemos, R., Saeed, A. and Anderson, T. (1995). Analysing safety requirements for process control systems. *IEEE Software* **12:3**, 42–53.

International Electrotechnical Commission. (1992). *IEC/SC65A: Functional Safety of Electrical/ Electronic/ Programmable Electronic Systems. Generic Aspects.* IEC (Secretariat) 123.

Karwowski, W. and Mital, A. (1986). Potential applications of fuzzy sets in industrial safety engineering. *Fuzzy Sets and Systems* **19,** 105–120.

Keller, A. Z. and Kara–Zaitri. (1989). Further application of fuzzy logic to reliability assessment and safety analysis. *Micro Reliability* **29:3,** 399–404.

Saeed, A., de Lemos, R. and Anderson, T. (1994). An approach to the risk analysis of safety specifications. *Proc. of 9th Annual Conference on Computer Assurance (COMPASS'94)*. Gaithersburg, MD. 209–222.

Saeed, A., de Lemos R. and Anderson T. (1995). On the safety analysis of requirements specifications for safety–critical software. *ISA Transactions* **34:3**, 283–295.

Schmucker, K. J. (1984). *Fuzzy Sets, Natural Language Computations And Risk Analysis*, Computer Science Press.

Wang, J., Yang, J. B. and Sen, P. (1995). Safety analysis and synthesis using fuzzy set modelling and evidential reasoning. *Reliability Engineering and System Safety* **47**, 103–118.

Wang, J., Saeed, A. and de Lemos, R. (1996). *Subjective Safety Analysis of Safety Requirements Specifications.* Technical Report . Dept. of Computing Science. University of Newcastle upon Tyne. 1996. (to appear).

Yang, J. B. and Sen, P. (1994). A general multi–level evaluation process for hybrid MADM with uncertainty. *IEEE Transactions on Systems, Man and Cybernetics* **24:10,** 1458–1473.