16th International Conference on
Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes
8-11 September 2014, Varna, Bulgaria

IMPROVING THE RELIABILITY OF DECISION-SUPPORT SYSTEMS FOR NUCLEAR EMERGENCY MANAGEMENT USING SOFTWARE DIVERSITY METHODS

Tudor B. Ionescu[1], *Eckart Laurien*[1], *Walter Scheuermann*[1]

[1] Institute for Nuclear Technology and Energy Systems, University of Stuttgart, Stuttgart, Germany

Abstract: Decision-support systems for nuclear emergency management (DSNE) are currently used to assist decision makers around the world in taking emergency response countermeasures in case of accidental releases of radioactive materials into the atmosphere. The present work has been motivated by the fact that, up until now, DSNE systems and the underlying atmospheric dispersion simulation codes have not been regarded as safety critical software systems and have therefore not been treated as such during the software testing and verification phase. The main goal of the current work is to improve the reliability of DSNE systems by adapting well-established methods from the domain of software reliability engineering to the case of atmospheric dispersion simulation codes. The effectiveness of the approach has been assessed using the atmospheric dispersion forecasts of two test versions of the widely used RODOS system.

Key words: *software reliability, decision-support, nuclear emergency, simulation codes, RODOS.*

INTRODUCTION

Decision-support systems for nuclear emergency management (DSNE systems) are in use in many countries producing nuclear energy (see for example (Galmarini, et al., 2004) for a comprehensive list). In the case of a significant release of radioactive materials into the atmosphere, the decision upon which sectors of the monitored area to evacuate first is taken based on atmospheric dispersion forecasts produced by DSNE systems. Erroneous dispersion simulation results could thus lead to false assumptions about the dispersion of radioactive pollutants and may thus bias evacuation priorities, which could endanger the health of the population residing in the areas affected by radiation. For these reasons, DSNE systems must be regarded as *safety critical* and their reliability as software-based systems must be improved using specific methods from the field of software reliability engineering (Vouk, 1993).

Drawing from the field of software reliability engineering, the current approach helps improve the reliability of DSNE systems by allowing the effective automated continuous verification of the underlying computer codes, which implement atmospheric dispersion models. The proposed approach is based on adaptations of the N-Version Programming (NVP) (Avizienis, 1985) and Recovery Blocks (RB) (Anderson & Kerr, 1976) methods to the case of atmospheric dispersion simulation codes (Ionescu, 2013). In safety-critical systems, these methods make use of several functionally-identical versions of a program to the end of increasing the trustworthiness of the results that the program is meant to produce. The rationale behind this approach is that software build differently is more likely to fail differently. In other words, it is conjectured that if several functionally identical versions of a program are being developed by independent organizations or teams, the likelihood for all of them to fail on the basis of common cause errors is lower compared to the case in which only one version of the program is used. The worldwide availability of over 30 atmospheric dispersion simulation codes justifies an approach to improving their reliability based on software diversity methods (Avizienis & Laprie, 1986). Functionally redundant simulation codes have already been integrated into DSNE systems such as the ENSEMBLE (Galmarini, et al., 2004) and RODOS (Ehrhardt, 1997) systems, thus providing the essential prerequisites for applying the N-Version Programming and Recovery Blocks methods. In order to effectively apply the NVP and RB methods to the case of DSNE systems, a taxonomy-based voter and an acceptance test specially-designed to work with atmospheric dispersion simulation results have been developed. R implementations of these methods can be downloaded from: https://clustio.googlecode.com/files/KS_Taxonomy_R_Scripts.pdf

METHODS

The recovery block (RB) method works as follows: given multiple functionally redundant versions of a computer program, the primary program version is executed first and the result of its execution is subjected to an application-specific acceptance test. If the result passes the acceptance test it becomes the block's output. Otherwise, the block reverts to the previous state and the next alternate is executed using the same inputs. This operation is repeated until the result of one of the program versions passes the acceptance test. The alternate versions may be either functionally equivalent with respect to the primary version or sacrifice accuracy and performance for the sake of reliability (Vouk, 1993). If none of the versions is able to produce an acceptable result, the block throws an exception.

N-version programming (NVP) (Avizienis, 1985) represents an adaptation to software of the broadly used N-modular redundancy fault tolerance method used in hardware systems, notably avionics. The NVP technique works as follows: two or more functionally identical program versions are executed in sequence or in parallel, whereby a generic selection algorithm is used to choose the correct result. The selection algorithm is usually a voter (or adjudicator) which compares the results of all program versions. If all results are equal (with a given tolerance), one of them is chosen randomly and becomes the module's output.

In order to apply the RB and NVP methods to the simulation codes of DSNE systems, it is necessary to develop an acceptance test and a voter specially tailored for atmospheric dispersion simulation results.

An Acceptance Test for Atmospheric Dispersion Simulation Results

The goal of the proposed acceptance test is to verify that the spatial distribution of substance concentrations and doses, as computed by some arbitrary dispersion code, is indeed Gaussian as suggested by theory (Etling, 2008). However, such an assessment is difficult to perform directly in an automated fashion since it would require a complex algorithm. Complexity is a strongly undesirable property for an acceptance test which needs to be much simpler than the program being verified (Anderson & Kerr, 1976). For simplicity, instead of verifying that the spatial distribution of the concentration (or dose) is Gaussian, the acceptance test will check that the distribution of the frequency function of the concentration or dose values obeys a certain hypothetical distribution. The frequency function is obtained by generating a histogram of the values for all cells of the discretized monitor area. The Weibull distribution (Weibull, 1951) can be applied to frequency functions derived from atmospheric radioactivity data (Apt, 1976). This represents the hypothesis of the proposed acceptance test, which will be performed on dispersion simulation results without regard to the spatial information. The Latitude-Longitude coordinates associated with each grid cell will be dropped from the two or three-dimensional matrices containing the concentration or dose values, which will be reduced to one-dimensional arrays representing the data sample upon which the Kolmogorov-Smirnov goodness of fit test (Massey, 1951) will be performed. The Weibull distribution is L-shaped, which makes the application of an automated goodness of fit test difficult (in an L-shaped histogram the vast majority of the measurements fall into the first interval from the left). To overcome this inconvenience, Weibull-distributed data samples can be transformed into exponentially distributed samples using a simple formula (Ionescu, 2013) before applying the goodness of fit test.

A Taxonomy-Based Voter for Atmospheric Dispersion Simulation Results

A taxonomy represents a classification into a hierarchy reflecting the relationships between the member elements of a given data set by considering the most relevant of its features (also called classification variables). In order to build taxonomies, it is necessary to first define a metric (or distance) upon the elements being classified or clustered. This allows for assessing the degree of similarity (or dissimilarity) between arbitrary elements with respect to the metric of choice. One way of obtaining taxonomies from arbitrary data is through hierarchical clustering (Hartigan, 1975). In the case of dispersion simulation results, a taxonomy of intra-model results can be obtained by relating new results to existing ones. By *intra-model* results it is meant that all results participating to the test are produced by the same dispersion simulation code. The taxonomy represents the memory (or history) of the dispersion code whereas the process of classifying a new result into an existing taxonomy may be regarded as a learning process. Dispersion simulation results are provided in the form of two or three-dimensional matrices whereby each matrix element corresponds to one area or volume element of the regular grid used to discretize the monitored area. The values contained in these matrices can be provided in arbitrary units. In the case of

radioactive trace species, the values corresponding to each grid cell can be one of integrated activities/doses or as activity/dose rates expressed in either [Bq] (Bequerel) / [Sv] (Sievert) or [Bq] / [Sv] per unit time.

From a mathematical point of view, atmospheric dispersion simulation results can be regarded as distributions of continuous variables in a finite space delimited by the monitoring area. The normalized version of the residual sum of squares metric can be used with the results of arbitrary dispersion codes since all of them produce matrix-based outputs. Applying a power transformation with a power $0<p<1$ will smoothen and level the data (i.e., very large values will become smaller and sub-unitary values will increase). Given a power $0<p<1$ and two dispersion simulation results r_1 and r_2 defined on an N by M cell monitored area for a number of TS time steps, the generalized RSS distance is given by:

$$GRSS(r_1, r_2) = \sqrt{\sum_{t=1}^{i \le TS} \sum_{i=1}^{i \le N} \sum_{j=1}^{j \le M} (r_1[t,i,j]^p - r_2[t,i,j]^p)^2} \quad (1)$$

The advantage of this metric is that it provides a high level of flexibility by letting the practitioner select the most suitable value of p. Experiments have shown that for the dispersion codes integrated in the RODOS DSNE system, setting $0<p\le0.4$ will make the wind direction be the determinant feature of the data. For $0.4<p\le1$, the determinant feature becomes the time and space-integrated dose or concentration in the considered area. Taxonomies can be constructed from a distance matrix computed using the GRSS metric by means of hierarchical clustering (Hartigan, 1975). Hierarchical clustering methods produce binary trees having the following properties: (1) each internal node is connected to exactly three other internal nodes or leaves and (2) leaves are connected to exactly one internal node.

Dispersion simulation codes fulfill the two basic requirements of the N-version programming paradigm: (1) they are developed and maintained by completely independent developer teams and (2) they implement dispersion models which all approximate the solution to the transport equation. Hence, they start from the same basic functional requirement: solving the transport equation numerically (Etling, 2008). However, dispersion simulation codes can pose considerable problems when it comes to comparing their results. Due to the inherent differences in the numerical schemes used for solving the transport equation, the results of the various dispersion codes will differ to such an extent that finding suitable thresholds and tolerance values to be used by voters can be rather difficult. Directly comparing results from codes implementing different dispersion models may also be biased towards reflecting inherent differences between the models rather than differences caused by residual software faults. Comparing taxonomies of intra-model dispersion simulation results instead of individual results therefore allows for the development of voting algorithms and the application of the N-version programming methodology to DSNE systems. That is because comparing taxonomies of results obtained through hierarchical clustering yields integer values, as opposed to comparing individual results.

We conjecture that, if two simulation codes correctly implement the same or different atmospheric dispersion models, then the taxonomies of results generated using the two simulation codes must be identical. A dispersion code is considered to correctly implement a numerical scheme for solving the transport equation if and only if it conserves its mathematical properties (i.e., determinism, strict monotonicity, and continuity). Failing to do so is taken to be an indication of the presence of residual software faults in the code caused by the misinterpretation of the model or some other programming errors. The same principle applies to post-processing steps, such as the gamma submersion and effective dose calculation. This conjecture represents the main theoretical result of this paper, which justifies the development of a taxonomy-based voter for dispersion simulation results.

EXPERIMENTAL VALIDATION AND RESULTS

In order to validate the acceptance test and the taxonomy-based voter for dispersion simulation results, two versions of the RODOS decision-support system (Ehrhardt, 1997) have been used, denoted RODOS_v1 and RODOS_v2. The operational RODOS system used by the authorities of several European countries offers a choice of three different dispersion simulation codes: RIMPUFF (a Puff model), DIPCOT (a Lagrangian particle model), and ATSTEP (a Puff model). The first version of the RODOS system was initially made available to the authors for the purpose of comparing its results with the ones produced by another DSNE system. This analysis revealed a number of flaws in the results produced by the three

dispersion codes integrated into the RODOS system. On the basis of these results the developers of the RODOS system admitted to have identified a fault in the ATSTEP dispersion code and carried out improvement works, which concerned DIPCOT and RIMPUFF as well. These works eventually lead to a new release of the system (i.e., RODOS_v2) which was made available to the authors one year later. Using the two versions of the system the same set of simulations has been carried out. These simulations were performed using two input case ensembles of 24 results each. The main requirements to the first ensemble (called the reference input case ensemble) were the following: (1) It shall cover the entire monitored area surrounding the chosen point of emission; (2) Input cases shall be defined such that from one case to another all of the following input parameters are varied and variation shall be equidistant: wind direction, wind speed, accident category, and diffusion category; (3) The wind direction shall be varied such that neighboring plumes overlap to some extent; (4) No precipitation and a point of emission with a flat surrounding topography shall be considered in order to reduce the number of influence factors to a manageable one. The names of the cases are coded as follows (see Figure 1): the first letter represents the diffusion category (A to F), the next three digits represent the wind direction, the number following the letter 'S' stands for the wind speed in m/s, and the number following the letter 'F' represents the release (or accident) category according to the DSRA risk study (Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit, 2004). The point of emission was chosen to be the KKP-2 (Philippsburg 2) reactor in Germany and the radius of the monitored area 16 km. Since in most of the 8 accident categories the entire inventory of the reactor is released within the first two hours from the emission start, the simulation duration was chosen to be 2 hours divided into 12 time steps of 10 minutes each, whereby the gamma radiation dose at ground level was considered. The second input ensemble was obtained using rotating wind fields.

For each version of the RODOS system the KS test was first run on the basis of the two input case ensembles. The results of these tests are summarized in Table 1 (figures in bold face indicate an improvement from one version of the simulation code to another). They show that the KS test is sensitive to the improvements brought to the RODOS_v2 system. The only exception is produced by the DIPCOT simulation code for the reference input case ensemble. In this case, the developers might have introduced a new fault while attempting to correct others—a risk that any software update entails. Notably the net improvement of ATSTEP_v2 over ATSTEP_v1 reflects the efforts of the developers aimed at removing the flaws revealed by our previous verification analysis of the system.

Table 1. Kolmogorov-Smirnov test results for the two input case ensembles for RODOS_v1 and RODOS_v2.

Reference Input Case Ensemble						
α = 1%	**ATSTEP_v1**	**ATSTEP_v2**	**DIPCOT_v1**	**DIPCOT_v2**	**RIMPUFF_v1**	**RIMPUFF_v2**
Passed	16 / 24	22 / 24	20 / 24	19 / 24	21 / 24	21 / 24
Percent	66.67%	**91.67%**	83.33%	79.17%	87.50%	87.50%
Second Input Case Ensemble						
Passed	17 / 24	20 / 24	15 / 24	19 / 24	20 / 24	23 / 24
Percent	70.83%	**83.33%**	62.50%	**79.17%**	83.33%	**95.83%**

Figure 1 shows the taxonomic trees corresponding to the three simulation codes integrated into the first version of the RODOS system, i.e., RODOS_v1. The pairwise Robinson-Foulds symmetric differences (Robinson & Foulds, 1981), denoted RF, which counts the number of branches present in one tree but not in the other one and vice-versa, computed for these trees were as follows (values in brackets represent percentages from the maximum RF distance for *N=24*): *AD_v1 = RF(ATSTEP_v1, DIPCOT_v1) = 16 (38%)*, *AR_v1 = RF (ATSTEP_v1, RIMPUFF_v1) = 16 (38%)*, and *DR_v1 = RF (DIPCOT_v1, RIMPUFF_v1) = 4 (9.52%)*. The GRSS distance with *p=0.6* was used to compute the distance matrix. The fact that the three taxonomies are not identical may be an indication that one or several of the underlying simulation codes contain one or several software faults. The RF distances between the taxonomies of results obtained by means of the second version of the RODOS system (RODOS_v2) are as follows: *AD_v2 = RF(ATSTEP_v2, DIPCOT_v2) = 4 (9.52%), AR = RF(ATSTEP_v2, RIMPUFF_v2) = 6 (13.63%)*, and *DR_v2 = RF(DIPCOT_v2, RIMPUFF_v2) = 2 (4.54%)*. The absolute improvement over RODOS_v1 is 28.48% for the pair ATSTEP-DIPCOT, 24.37% for ATSTEP-RIMPUFF, and 4.98% for DIPCOT-RIMPUFF, which reflects the improvements brought to the simulation codes of the RODOS system.

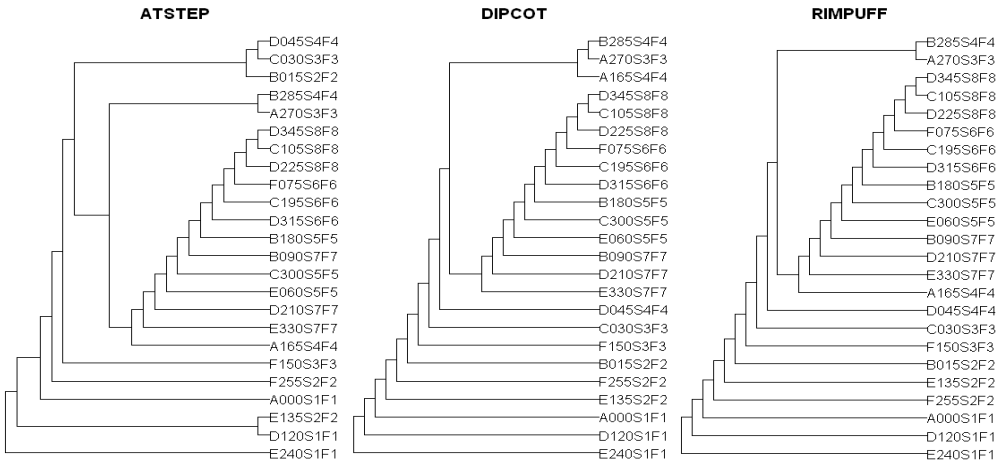

Figure 1. Taxonomic trees corresponding to the simulation codes integrated in RODOS_v1. The trees were constructed by means of the GRSS metric with *p=0.6*. The centroid cluster joining criterion was used.

CONCLUSION

In this paper we have argued that DSNE systems are safety-critical and therefore should be treated as such from a software reliability point of view. The results of the experimental evaluation of the methods introduced in this work proved that the Kolmogorov-Smirnov acceptance tests and the taxonomy-based voter are sensitive to the improvements brought to the dispersion simulation codes of the RODOS system. Therefore, the proposed acceptance test and the voter can be used as software reliability metrics, which can help find software faults in dispersion simulation codes throughout the useful life of any DSNE system.

REFERENCES

Anderson, T., & Kerr, R. (1976). *Recovery blocks in action: A system supporting high reliability*. Proceedings of the 2nd international conference on Software engineering (pp. 447-457). Los Alamitos: IEEE Computer Society Press.

Apt, K. E. (1976). Applicability of the Weibull distribution to atmospheric radioactivity data. *Atmospheric Envir. 10*, 777-782.

Avizienis, A. (1985). The N-Version Approach to Fault-Tolerant Software. *IEEE Trans. Softw. Eng. 11*, 1491-1501.

Avizienis, A., & Laprie, J. C. (1986). Dependable computing: From concepts to design diversity. *Proceedings of the IEEE 74*, 629-638.

Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit. (2004). Leitfaden für den Fachberater Strahlenschutz der Katastrophenschutzleitung bei kerntechnischen Notfällen. München-Jena: Elsevier Urban & Fischer.

Ehrhardt, J. (1997). The RODOS System: Decision Support for Off-site Emergency Management in Europe. *Radiation Protection Dosimetry 73*, 35-40.

Etling, D. (2008). *Theoretische Meteorologie: Eine Einführung*. Heidelberg: Springer.

Galmarini, S., Bianconi, R., Klug, W., Mikkelsen, T., Addis, R., Andronopoulos, S., & Astrup, P. (2004). Ensemble dispersion forecasting, Part I: concept, approach and indicators. *Atmospheric Environment 38*, 4607-4617.

Hartigan, J. A. (1975). *Clustering algorithms*. New York : John Wiley & Sons.

Ionescu, T. B. (2013). *Reliability of Decision-Support Systems for Nuclear Emergency Management*, OPUS Uni Stuttgart (dissertation): http://elib.uni-stuttgart.de/opus/volltexte/2013/8627/

Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association 46*, 68-78.

Robinson, D. R., & Foulds, L. R. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences 53*, 131-147.

Vouk, M. A. (1993). *Software Reliability Engineering*. In A. Kent, & J. G. Williams, Encyclopedia of Microcomputers (pp. 161-178). New York: Marcell Dekker

Weibull, W. (1951). A statistical distribution function of wide applicability. *J. Appl. Mech.-Trans. ASME 18*, 293–297.