Identifying Software Hazards with a Modified CHAZOP

Bernhard Hulin
Deutsche Bahn AG
DB Systemtechnik
Munich, Germany
E-mail: bernhard.hulin@deutschebahn.com

Rolf Tschachtli
Deutsche Bahn AG
DB Fahrzeuginstandhaltung
Munich, Germany
E-mail: rolf.tschachtli@deutschebahn.com

Abstract—**CHAZOP is one of the most popular methods for identifying hazards of software. However, the classical HAZOP methodology as well as the CHAZOP methodology has four technical insufficiencies when applied to software: Ambiguity, incompleteness, nonsensicality and redundancy of HAZOP expressions. This present paper shows a modification of CHAZOP to overcome these insufficiencies. The reasons for these insufficiencies are a non-specified HAZOP language and missing guide words. We therefore, define a HAZOP language and identify missing guide words. The definition of the language is based on the items: Actions, objects, and their attributes. In contrast to the classical HAZOP, the modification defines rules for combining these items with guide words. One of the key ideas of the language is to use HAZOP parameters twice whenever possible: As objects and as attributes. In practice, this means that an attribute is additionally analyzed as if it were a software variable. We call this concept manifestation since in our new method attributes are also manifested in variables. For evaluation, the modified method is compared with the traditional one with the example of a safety-relevant software-controlled system using the windows registry. By means of this example, it is shown that more hazards can be found with the modified CHAZOP than with traditional method.**

Keywords - HAZOP; deviation; parameter-manifestation; hazards.

I. INTRODUCTION

HAZOP is one of the most widely used techniques for the identification of hazards in the production and operation of technical systems. Coming from the chemical industry [1] [2] originally, HAZOP has been adapted to other industrial areas. Later it was adapted for different business areas such as Computers [3][4][5], where it is known as CHAZOP.

The idea behind HAZOP is to combine parameters with guide words to gain indicators of possible failures of a system (we call these combinations HAZOP expressions), then to formulate interpretations of these HAZOP expressions within a team, to extract deviations of the set of interpretations (since a few interpretations are not deviations but are desired), and finally to extract hazards out of the set of deviations [15][16]. HAZOP parameters can be system components, their attributes, as well as actions and their attributes.

We use HAZOP and CHAZOP, respectively, for the identification of hazards induced by the software of railway vehicles. We use these hazards as inputs for the risk assessment [11][12], resulting in a SwSIL classification [17]. These SwSIL classifications are demanded by law for new or modified software used in railway vehicles. They are to be performed by assessors accredited by the German Federal Railway Authority.

Although HAZOP is the most widely used technique for the identification of hazards it has three drawbacks as mentioned by several authors: Amount of time, high costs and safety-gaps. For reducing costs and saving time several authors suggest an electronic system for the management of deviations [7][13]. However, even if the management does not cost any time at all, the expenditure of time for HAZOP meetings remains almost equally high. Our experience in software assessment for railway vehicles is that HAZOP meetings for SwSIL classification for one railway component last about 1.5 days with an average of 4 participants thus an average effort of 6 man-days is spent for this step. Depending on the application, we integrate persons covering the following roles: Operator (or user), maintenance manager, project manager, rollout manager and software developer. Sometimes it is necessary to integrate other roles such as experts for other relevant aspects like fire resistance or EMC or experts that have knowledge about interacting components. Based on our experience, the minimum duration for HAZOP of system modifications is about 4 days.

The efficiency of the meeting can be increased on the one hand by building more meaningful HAZOP expressions, which do not need to be interpreted, and on the other hand by not generating useless HAZOP expressions. These two problems can also be found in literature (see [5] p. 55, [6] pp. 73, 74 and [7] p. 68), but a solution for them has not been shown in literature. Moreover, we found in our meetings that ambiguities of HAZOP expressions can lead to missing

 ISBN: 978-1-61208-132-8

hazards, since there is no proof that all possible interpretations of an ambiguity have been observed.

To overcome these insufficiencies, we modified the generation of expressions. With this modification it is possible to generate meaningful and unambiguous HAZOP expressions manually or alternatively automatically by software. Each HAZOP expression then corresponds to exactly one deviation.

With this modification it is possible to find more deviations and thus more hazards than with the classical generation of deviations, and the workload can be transferred from meetings to the office, which saves manpower. In contrast to earlier publications [10], this paper enhances the modification with concepts necessary for hazard identification in software. The core of this novel concept is developed within Tschachtli's master thesis [14] and considers each HAZOP parameter twice: First as a traditional HAZOP parameter and second as a model expressed in software. For attributes such as pressure or velocity it is a manifestation into an object.

The decision to modify HAZOP instead of developing a new technique or modifying another technique was based on the fact that we haven't found any other method essentially different to HAZOP that performs the identification of hazards in such a structured way.

The paper first describes the traditional method as well as its insufficiencies. Then these insufficiencies are examined in detail. From this examination activities are conducted. They result in new guide words and a new procedure of HAZOP analyzes. We conclude with an application of this modified method.

II. RECENT METHOD

As the initial input for HAZOP we use our knowledge, descriptions of the component, and checklists of former assessments. The descriptions should describe functions of the component, interfaces to other components, its input and output, the internal structure (or architecture), operational conditions, and maintenance modes. Operational conditions should answer at least the questions where, when, how, by whom, and how often the component is used. Information about the output and the interfaces is important to evaluate the effect on other components. For example, a component which does not have any safety-related functions can be cabled to a vehicle bus, which is used for the transfer of safety-related data. For chemical plants, an overview of aspects that have to be contained in a description can be found in [7].

With the use of checklists of former assessments, some HAZOP expressions do not need to be discussed in detail or at all in a meeting. HAZOP expressions are classically generated by combining HAZOP guide words pair-wise with HAZOP parameters. As HAZOP parameters we use objects, attributes of objects and actions (e.g., system functions or user operations). The HAZOP parameters are extracted from documents (descriptions and former assessments), knowledge and discussion. Each HAZOP expression is then analyzed under environmental conditions and operational states.

The set of objects consists of system components, such as trains; subcomponents; subjects, such as humans; and environmental objects, such as tunnels and bridges. Subjects can be members of special groups of persons such as train-drivers, conductors, passengers, disabled persons, or children.

Attributes are attributes of these objects. For example, they contain electrical current, velocity of the train, contrast of the display and so on.

Actions are taken of actions of objects and actions of humans. The main actions, which are always taken into consideration in our assessments, are safety-relevant actions that can be performed by a train or the train driver.

Then the HAZOP expressions are generated and are considered within different scenarios. The set of scenarios contains each operational mode, such as passenger transportation, cargo transportation, cleaning mode, maintenance mode, test mode and so on, locations where this train can be, such as Germany, Austria, France, tunnels, bridges, elevation of track, radius of curves and so on, and hazard-modes of the train, such as onboard-fire, onboard smoke, loudness of noise, and failure of different components.

Weather conditions, such as temperature, fog, rain, snow, and so on are only taken into account as far as the software has to react on it. This is normally the case in displaying software [8] where the contrast may not be enough or on air conditioning software.

A very important analyzes is the change of scenarios where the train transits from one to another scenario. This is for example often the case on European country borders where nearly each country has its own electrical current and train control system.

III. INSUFFICIENCIES OF HAZOP

There are four methodological and technical insufficiencies of HAZOP with respect to its HAZOP expressions. These are ambiguity, incompleteness, nonsensicality, and redundancy of the HAZOP expressions. They result from the assignment of HAZOP expressions to interpretations. Therefore, exactly the four mentioned insufficiencies exist (see Figure 1).

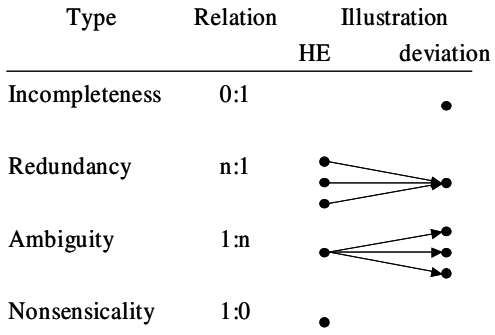

Figure 1. Insufficiencies of HAZOP. How many HAZOP expressions (HE) can be associated with how many deviations.

The danger in ambiguity and incompleteness of HAZOP expressions are missing hazards and thus reduced safety. Nonsensicality and redundancy of HAZOP expressions just results in meetings being disrupted and lasting longer. Therefore, obviation of redundancies and nonsensical HAZOP expressions would improve the method, but this is not safety-critical. Of course, ambiguity also indirectly results in meeting lasting longer, since interpretations have to be found and discussed.

Although three of the four insufficiencies – ambiguity, nonsensicality and redundancy – are known in literature [5] (page 55), [6] (pages 73, 74), [7] (page 68), the problems have not yet been analyzed in depth and no corrective was suggested.

A. *Ambiguity*

Ambiguities arise because the context of the HAZOP expressions is interpretable. There is no one-to-one relation between a HAZOP expression and a deviation. For example the HAZOP expression “no pressure” with the HAZOP parameter “pressure” and the guide word “no” can be interpreted as “no pressure measurable”, “no pressure displayed”, “pressure = 0 Pa” or “there is no variable pressure within an ini-file”.

A second example is the HAZOP expression “more pressure”. Traditionally, this is interpreted as “more pressure than expected”, “more pressure than specified,” or “more pressure than outside the cylinder”. Even the interpretation “more pressure than expected” is ambiguous. It can either mean that the real pressure is higher than expected or that the software variable has a higher value – or both.

B. *Incompleteness*

With traditional HAZOP certain failures and thus hazards cannot or can barely be associated with any HAZOP expression. The conclusion is that the set of HAZOP expressions is not exhaustive and not detailed enough.

For example, each railway train has an identifier such as “de-484-22a-1”. The HAZOP expression “identifier other” can result in a hazard such as train not being reachable. I would be reasonable sure, however, that you have not identified the special deviation “identifier of train 1 is equal to identifier of train 2”. This could be critical if two trains are coupled or have to be dispatched at a central local display within the same district. Certainly, this is a very special case of the HAZOP deviation mentioned but it is very hard to build on this base.

However, the HAZOP expression “identifier other” includes the special case “identifier contains a blank”, too.

C. *Nonsensicality*

Nonsensical HAZOP expressions are also results of the arbitrary combination of each HAZOP parameter with each HAZOP guide word, without considering the context or reasonability of this connection. Examples for this kind of HAZOP expression are “tree early” or “name higher”. Nonsensical HAZOP expressions cost time and even nerves.

D. *Redundancy*

For the redundancies, the reason is similar to the nonsensical expressions. In some HAZOP expressions, the same statements can show up multiple times, because the meaning of the statement is identical. For example, the HAZOP expression “pressure other” includes both “pressure larger” and “pressure smaller”. If you have more than one surname, please make sure that the Volume Editor knows how you are to be listed in the author index.

IV. REASONS OF INSUFFICIENCIES

The main reason for the insufficiencies mentioned is that the HAZOP methodology should induce and support human thinking and interpreting. HAZOP is made for suggesting directions for human thinking with respect to deviations of a system. Therefore, overcoming the insufficiencies mentioned means limiting the degree of interpretation. The possibility to interpret HAZOP expressions is based on at least four aspects.

- Missing ambiguity differentiation for HAZOP parameters
- Unrestricted combination of guide words and HAZOP parameters
- Missing interrelations between HAZOP parameters
- Missing guide word

HAZOP parameters can be ambiguous on their own. One instance of this ambiguity is based on natural language. A prominent example is the data bus within a passenger bus, where “bus” has two meanings. The second kind of ambiguity is based in the dualism between real things and their model. For example the concentration of a gas is a real world attribute while the variable “concentration” within a software program is an object.

Nonsensicality and redundancy of HAZOP expressions are reasoned in unrestrictedly combining guide words and HAZOP parameters. A restriction in combinations, for example, can be that for certain HAZOP parameters the guide word “other” is used, whereas for others the guidewords “less” and “more” are used.

As we found in our HAZOP meeting, missing comparisons between two or more HAZOP parameters result in incompleteness of the set of hazards. Usually, HAZOP expressions are interpreted as a comparison with expectation, e.g. “larger than” “expected”. An analysis has to be done about which HAZOP parameters can be compared, and with which operators they can be compared. In contrast to [10], we widened this topic from comparison to relations.

One reason for undetected deviations is missing guide words. The challenge is to find missing guide words without introducing lots of new guide words.

V. CONCEPT FOR IMPROVING HAZOP

Our strategy for eliminating these reasons is first to differentiate HAZOP parameters and concretize their meanings. After that, rules for combining guide words and HAZOP parameters are generated. This step also focuses on interrelations between HAZOP parameters. Finally missing guide words are added. The idea here is to compare guide

 ISBN: 978-1-61208-132-8

words of HAZOP with the set of mathematical operators, and to add those operators, which are missing.

A. *Differentiation of HAZOP Parameters*

HAZOP parameters can be separated into three different kinds: Objects, actions, and attributes (see Figure 2). The set of objects consists of material and immaterial objects. Examples of material objects can be found in Section 4. Immaterial objects are for example source code (among other object with variables), files, and processes. Actions are discussed elsewhere [18].

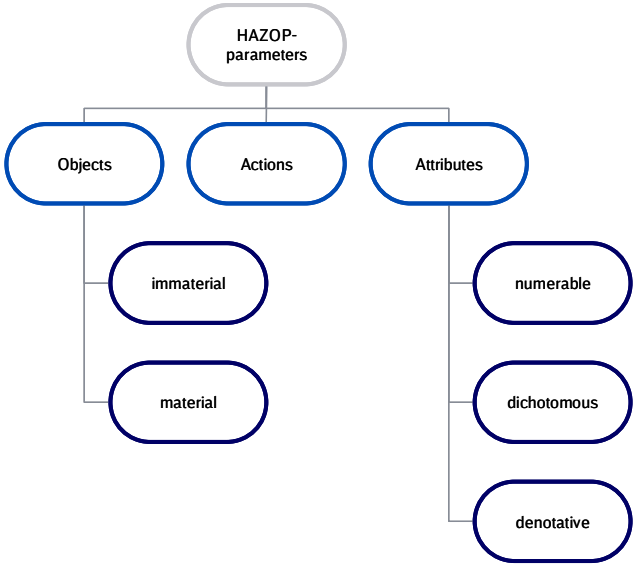

Figure 2. Types of HAZOP parameters.

Attributes belong to objects and can be of type numerable, dichotomous, and denotative. Examples for denoting attributes are names, telephone numbers, characteristic curves of sensitivity of a chip, or charts of shares. Numerable attributes with units for instance are the size in kilograms or liters, while numerable attributes like the amount or degree of capacity might only have a reference object or reference attributes. Dichotomous attributes are like the attribute of a CD – this can be only writable or not writable. We give each dichotomous attribute a name such that it can only have the values true or false.

Up to now, it is still unclear if a HAZOP parameter refers to an attribute or to an immaterial object – for instance, a software variable. For differentiation, we preface each HAZOP parameter an identifier that refers to the type – for example, "variable pressure" and "attribute pressure". A few immaterial objects of software refer to real attributes of objects. In the case of variables, a real attribute is modeled into a variable. This modeling of an attribute into an immaterial object – for instance, a variable – is termed manifestation by us.

Since we do not know a priori if a certain HAZOP parameter is just a real attribute or is implemented as a variable too, we assume in the beginning that each attribute is also implemented as a variable, and thus put them as objects into our list of objects. On the one hand, this induces a duplication of the amount of analyzable items, but on the other hand, this procedure reduces the probability of omitting deviations and thus hazards.

Some points about manifestation are worth noting. One of the most important things is that manifested objects are regular objects. They have attributes and consist of partition objects. In the case of variables, partition objects can be a data-type identifier, the name of variable, and the content of the variable. For example, in Figure 3 where the object "compression control system" is divided into the partition-object "pump" and "control software", the attribute pressure is manifested in a variable pressure.

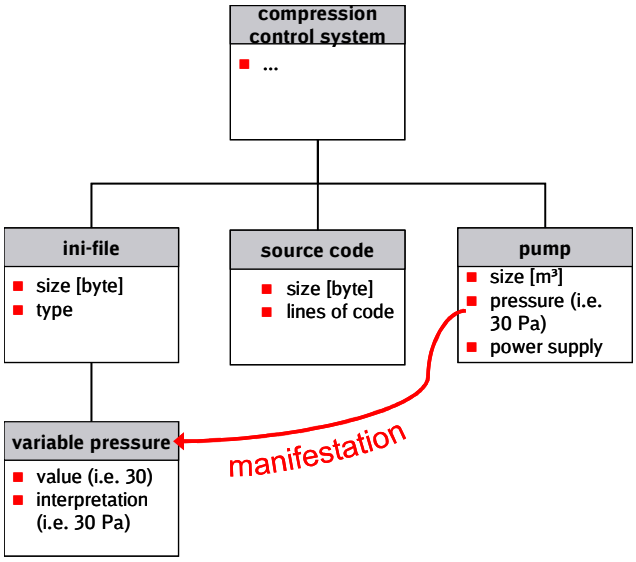

Figure 3. Example of manifestation: The attribute pressure is manifested in a variable pressure that is an object

Attributes of variables are for example, value and interpretation of value. For example, the value "30" of a variable pressure can have the interpretation "30 bar" or "30 Pa". The desired case is that the interpretation of a variable is equal to the real attribute. Note, that attributes of variables are sometimes manifested in other variables.

The second important thing is that attribute type and variable type are independent of each other. For example a numerable attribute is not necessarily a numerable variable but can be modeled in a string, too.

B. *Rules for Generating HAZOP Expressions*

In [10], we analyzed which types of interpretation of HAZOP expressions are in traditional HAZOP. For that, we set up a matrix with a column for each analyzed HAZOP parameter and a row for each HAZOP guide word. Clouds of guide words such as {no, none, not, never} and {reverse, inverse, opposite} were split. HAZOP guide words were collected from publications [1][2][3][4][5][6][9]. HAZOP parameters were taken from recent assessments as well as

 ISBN: 978-1-61208-132-8

from fictive examples. Each of the HAZOP parameters was differentiated with respect to its type. Resulting expressions were characterized by logical, structural, and mathematical aspects.

From that analysis we identified the following types of interpretations of HAZOP expressions in the traditional HAZOP.

1) statement of existence: This kind of expression states that an object does not exist or partially exists. The statement consists of an object and a quantifier or qualifier.

2) comparison of attribute to a special value: This type of statement is received by HAZOP expressions like "no pressure", "not nice" and "no country", which is interpreted as "pressure = 0", "niceness = false" and "country = {}".Type (2) and (3) can be identical but often are not. Special values are software specific values that can cause problems in each kind of software.

3) comparison of attribute to expectation: The difference to a comparison of to a special value is that the expectation is project-specific.

4) junction of two statements of existence: This type occurs if objects are used with the guide words "as well as" and "instead of". Examples are "bus A as well as something else", which can be interpreted as "bus A exists and something else exists".

5) junction of two comparisons of attributes: Here, the guide words "as well as" and "instead of" are used in combination with attributes. For example "beauty as well as something else" can be interpreted as "beauty = true AND another attribute is true ".

C. Completing Guide Words

As mentioned above, interrelations between HAZOP expressions are missing. There can theoretically be the following types.

- Relations between two attributes
- Relations between two objects
- Relations between an attribute and an object

For statements of existence we extend the set of guide words by "completely" and "multiple times". The extensions have the meaning "against expectation one object (already / still) (completely) exists" and "against expectation multiple objects of the same type exist". Sometimes it also makes sense to add "multiple times partially", which is to be interpreted as "against expectation multiple objects partially exist".

Guide words for comparisons of attributes with a special value, with expectation and with another attribute are "=" and "≠". For comparing numerable attributes "<" and ">" can be used, too. For dichotomous and denotative attributes "<" and ">" do not make sense. Moreover, in a few applications for numerable attributes the comparators "|" and "∤", which mean "divides" and "does not divide" can be useful as well. With these comparators one can verify if an attribute is multiple times another attribute or not.

Guide words for the interrelation of objects are "is (not) partition object of" or "is (not) contained in" or "contacts". A specialization of this case is the relation of an object to a predefined special object. This special case is very useful for the analysis of software variables. With the help of this specialization we are able to check a string for blanks, tabulators, quotation marks or other special characters which are often troublesome.

A good example for a relation between an attribute and an object is "manifested in".

The junctions mentioned of rule 4 and rule 5 are very narrow since they join a statement with "something else". They have to be generalized in such a way that two statements can be combined without limitation. However, this problem will be a topic of future papers.

VI. RESULTS

To put the new method into competition with the traditional HAZOP we chose one application for hazard identification where we applied the traditional method and afterwards the new modified method. This order of applying the methods was chosen since we believe that the new method is more complete than the traditional one.

As the application we used a system for displaying the electronic schedule with speed information (including speed reduction intervals) to the train driver. Information displayed to the train driver is in most cases safety-critical.

Some configurations of this system are configured in configuration files like the Windows Registry, ini-files, and so on. Consequently, false configurations within these files can affect information displayed. For an overview of possible failures, see [8].

Entries of the registry are modified remotely by wireless data exchange via the mobile network. Normally this happens if a new version of the system is remotely installed or a new functionality is to be enabled.

Our task is to rate the entries of the registry file according to their safety impacts. Thus, for each entry we have to identify the hazards, the probability of occurrence, the severity of the consequences, the probability of detection in case of occurrence and the chance to escape the critical situation. The result of this analysis is a safety integrity level [19].

With the new method in relation to the conventional HAZOP we have identified the following additional hazards:

- Registry value can be of greater size than the available hard disk space
- Values do not exist
- The data type of a value can change via a software update
- Wrong upper and lower cases inside the value data or the value name

The first two hazards are coped by checking for the completeness of the transferred files and entries and default values set in the program. The third hazard was fought by the cross check and constraining the SIL classification. The constraint is that each change of type has to be reassessed.

 ISBN: 978-1-61208-132-8

Upper and lower cases are not a problem since the program is case-insensitive.

Therefore the SIL classification by the traditional method does not have to be changed. The consecution of SIL can be understood as a good quality of our former work. On the other hand the additional hazard identified is an argument for the quality of the new method.

VII. CONCLUSION

In this paper, we showed an approach to overcome the insufficiencies of hazard identification with HAZOP in the area of software. We introduced the concept of manifestation and added missing types of deviations by adding interrelations and completing the set of operators. The new concept, presented in this paper can eliminate the insufficiencies of HAZOP. We showed the improvement with a railway example.

Although we just described one application of the modified HAZOP our method has been prooven as good and practicable in other projects, too – such as the SwSIL classification of a power-transformation unit and a diagnostic unit for detecting wheel defects. Of course our method also has some limitations. The limitations are that in the preparation phase it is nearly impossible for one person to figure out each important attribute of an object. This is explained by the fact that each object has infinitely many attributes. This limitation is not a special feature of our method but adopted by HAZOP. Thus this issue is not worse in our method than in HAZOP.

Further steps of our work are adding functions, actions, and events to our concept. The guide word application is probably different for them. Furthermore, combinations of expressions have to be analyzed. Moreover, the new method also has to be evaluated with more applications with respect to time consumption.

ACKNOWLEDGMENT

We want to thank Jenny Schulze for her excellent input during her time as master student, and Dr. Dirk Leinhos for enabling and supporting the progress of improving our work as assessors.

REFERENCES

[1] Chemical Industries Association, “A Guide to Hazard and Operability Studies”, 1977.

[2] T. A. Kletz, “Hazop and Hazan: Identifying and Assessing Process Industry Hazards”, 4th edition, Institution of Chemical Engineers, Rugby, UK, 1999.

[3] J. Love, “Process Automation Handbook – A Guide to theory and practice”, Springer Verlag, Berlin, 2007.

[4] S. Mannan, “Lees loss prevention in process industries – hazard identification, assessment and control”, Vol. 1, Elsevier, 2004.

[5] T. Kletz, P. Chung, E. Broomfield, and C. Shen-Orr, “Computer Control and Human Error”, Instn.of Chem.Enginrs, 1995.

[6] F. Redmill, M. Chudleigh, and J. Catmur, “System Safety – HAZOP and Software HAZOP”, Wiley and Sons Ltd., Chichester, U.K, 1999.

[7] I. Faisal, F. Khan, and S. A. Abbasi, “Towards automation of HAZOP with a new tool EXPERTOP”, Environmental Modelling & Software, no 15, Elsevier, pp. 67-77, 2000.

[8] B. Hulin and T. Schulze, “Failure analysis of software for displaying safety-relevant information”, in Reliability, Risk and Safety: Theory and Applications, Taylor and Francis Group, pp. 1327-1331, 2010.

[9] M. Rausand, “HAZOP - Hazard and Operability Study”, Norwegian University of Science and Technology, 2005, www.caia.co.za/files/Hazop_Technique_MarvinRausand.pdf, last access 27th January 2011.

[10] B. Hulin and R. Tschachtli, “Generating unambiguous and more complete HAZOP expressions”, in Reliability, Risk and Safety, Taylor & Francis Group, London, pp. 1-7, 2010.

[11] M. Geisler, „Betriebliche und technische Risiken managen“, Deine Bahn, 10, 2010, pp. 9-14.

[12] B. Milius, “A new classification for risk assessment methods”, in Proceedings of 6th Symposium FORMS/FORMAT 2007, Jan. 2007, pp. 258 – 267.

[13] S. A. McCoy, S. J. Wakeman, F. D. Larkin, M. L. Jefferson, P. W. H. Chung1, A. G. Rushton, F. P. Lees, and P. M. Heino, “HAZID, A Computer Aid for Hazard Identification: 1. The Stophaz Package and the Hazid Code: An Overview, the Issues and the Structure”, Process Safety and Environmental Protection, Volume 77, Issue 6, Nov. 1999, pp. 317-327.

[14] R. Tschachtli, „Entwurf einer Methode zur Identifikation von Fehlermöglichkeiten bei der Entwicklung von Softwarekomponenten“, Master Thesis, Fachhochschule Bingen, Oct. 2009.

[15] Ministry of Defence, “HAZOP Studies on Systems Containing Programmable Electronics”, Defence Standard 00-58, Issue 2, Part 1 and 2, May 2000.

[16] International Electrotechnical Commission, “Hazard and operability studies”, BS IEC 61882, Aug. 2001

[17] CENELEC European Committee for Electrotechnical Standardization, “Railway applications – Communications, signalling and procession systems – Software for railway control and protection systems”, EN50128, Nov. 2000.

[18] J. Schulze, “Improvement of Hazard Identification in Railway Software”, Master Thesis, Chalmers University of Technology, Sep. 2010.

[19] International Electrotechnical Commission, “Functional safety of electrical/electronic/programmable electronic safety-related systems”, IEC 61508, Edition 2.0, Apr. 2010.

 ISBN: 978-1-61208-132-8