Application of Case-Based Reasoning to the safety assessment of critical software used in rail transport

Habib Hadj-Mabrouk

Application of Case-Based Reasoning to the safety assessment of critical software used in rail transport

Habib Hadj-Mabrouk

University Gustave Eiffel, IFSTTAR, Scientific Direction,
14-20 Boulevard Newton, F-77447 Marne la Vallée, France
E-mail: habib.hadj-mabrouk@univ-eiffel.fr

Application of Case-Based Reasoning to the safety assessment of critical software used in rail transport

.....
.....
.....
.....

Abstract. The risk assessment of railway accidents requires the implementation of several safety analysis methods such as Preliminary Hazard Analysis, Functional Safety Analysis, Analysis of Failure Modes, their Effects and their Criticality and Software Error Effect Analysis (SEEA). The study proposed within the framework of this article concerns the SEEA method whose objective is to determine the nature and severity of the consequences of software failures, to propose measures to detect errors and improve the robustness of the software. The goal is to develop a new approach to analysis and evaluation of the safety of critical software, based on machine learning and more precisely on the Case-Based Reasoning (CBR). The approach adopted involves two main activities: The first step in acquiring safety knowledge consists in extracting, modeling and archiving dangerous situations to produce a library of standard cases which covers the entire problem. The second stage of machine learning is to exploit historical knowledge (experience feedback) in order to assist safety experts in their critical task of analyzing and assessing the safety of software involved in guided or automated rail transport systems. This second activity involves the use of Case-Based Reasoning (CBR).

Keywords: Railway transport, Case-Based Reasoning, Safety, Software Error Effect Analysis, Assessment, Accident prevention

1 Introduction

The basic principle of CBR is to deal with a new problem by remembering similar experiences which have occurred in the past. The objective of the study is the development of a case-based reasoning (CBR) system to help safety experts judge the completeness and consistency of Software Error Effect Analysis (SEEA). The purpose is to exploit historical SEEAs, which have already been carried out on approved safety-critical software, in order to assess SEEA of new software. Very schematically, the objective of this study is to exploit a case base formed by historical SEEA (source case), carried out on already validated and certified software, in order to explain or evaluate a new case of SEEA on new software (target case) and therefore help and stimulate the imagination of experts in the field in the search for new critical situations contrary to safety that requires the implementation of safety barriers or instructions and adequate preventive measures.
This article is organized around seven major paragraphs. The first paragraph presents the main methods of railway safety analysis and in particular the SEEA method which is the subject of this manuscript. The objective of our study as well as the approach adopted for the development of an aid tool for the analysis and evaluation of the knowledge involved in the SEEA method are detailed in the second paragraph. We demonstrate that the chosen approach requires the use of AI techniques and in particular the joint use of conventional knowledge acquisition approaches and more formal methods of automatic learning. The third paragraph is devoted to an analysis of the literature on AI techniques; emphasis is placed on the CBR. This same paragraph presents several examples of CBR applications for rail transport safety. This bibliographic study enabled us to position, in paragraph four, our contribution with respect to the state of the art. The fifth paragraph finally proposes a new method of assessment of critical software safety based on the CBR. In order to demonstrate the feasibility and appropriateness of the proposed method, the sixth paragraph presents an example of application which is based on 224 SEEA cases from the knowledge acquisition phase of already certified rail transport systems and commissioning in France (experience feedback). The results obtained are presented in the last paragraph.

The harmful consequences of rail accidents, which sometimes lead to loss of life and the destruction of the system and its environment, are at the basis of the implementation of a REX (experience feedback) system considered as the essential means to promote the improvement of safety. The main objective of process REX is to learn from an experience lived to avoid its reproduction. For a railway operator (railway undertaking or infrastructure manager), the objective of the REX is to improve the level of safety of its operation by taking advantage of negative past events (accidents, serious incidents, near misses, etc.) or positive (good practice, benchmark, etc.). The REX aims not only to reduce in number and / or severity, the malfunctions of the system (men, installations, procedures, environment), but also the implementation of the most effective measures to control the risks related to the life cycle of railway systems (design, construction, operation, maintenance).

In the context of this study, it seems important to us to clarify the term REX (experience feedback). It would not be a feedback on the use and exploitation of data on rail accidents and incidents, but rather data from existing safety files relating to SEEA of two rail transport systems put into service in France. In this context, the REX concerns the data carried out, from the design phase, by the system builders and safety experts.
At this stage of the critical software safety analysis (design phase), some of the assumptions developed by the SEEAs can lead to incidents or accidents. However, the use of other complementary safety assessment methods (see Figure 1), the implementation of a certification process (carried out by an independent body) and finally the authorization procedure for setting circulation of the system carried out by the national safety authority (EPSF in France) can mitigate or avoid the occurrence of undesirable situations contrary to safety.

2 Safety software analysis and assessment methods

In order to guarantee an acceptable level of risk vis-à-vis humans, the system and its environment, safety experts use several methods of safety analysis shown in Figure 1: Preliminary Hazard Analysis (PHA), Functional Safety Analysis (FSA), Analysis of Failure Modes, their Effects and their Criticality (AFMEC), Software Error Effect Analysis (SEEA), Fault Tree Analysis (FTA), etc. The PHA aims to identify potential accidents related to the transport system and its interfaces in order to evaluate them and propose solutions to remove reduce or control them. The FSA aims to justify that the design architecture of the system is safe against potential accidents identified by the PHA and therefore to ensure that all safety provisions are taken into account for cover potential hazards or accidents. These analyzes provide safety criteria for the design of the system and the realization of hardware and software safety equipment.
The hardware safety analysis focuses on electronic boards and interfaces defined of safety. This study implements two types of analysis: 1) An “inductive” analysis by analysis of failure modes, their effects and their criticality (AFMEC) and 2) A “deductive” type of analysis by searching for scenarios that run counter to safety and that make it impossible to comply with the safety criteria derived from the FSA. This analysis usually requires FTA method.
Finally at the software level, there are many methods contributing to the development and evaluation of software [1]:

- Static analysis methods (SEEA: Software Error Effect Analysis, BDD: Boolean Decision Diagrams, Detailed review, Metrics, etc.)
- Specification methods: Petri network, SADT (structured analyzes technical design), etc.
- Development methods: Languages of algebraic specifications (Life, Act-one, CLEAR, OBJ, etc.), Formal methods (Method B, Method Z, VDM: Vienna Definition Method, etc.), N-versions, Overlay blocks, etc.
- Version managers (SCCS: Source Code Control System, RCS: Revision Control System, CVS: Concurrent Versions System).

- Dynamics tests (Code exploration, Non-regression tests, etc.)
- Other techniques: Proof of programs, Avoidance of faults (in order to prevent the occurrence of errors), Tolerance of faults (in order to preserve the service despite the occurrence of faults), Elimination of faults (Software tests, Critical review code), etc.

To complete this process, we can also add the "Human" component which actively participates in the generation of railway accidents and incidents. Indeed, the human operator is a paradoxical element: in situations of stress or fatigue, it can be an element of the loss of the reliability of a system. However, in certain critical situations of insecurity, it can be a factor of reliability, by restoring the proper functioning of the system, sometimes by actions not provided for by the operating safety regulations, but, linked to its knowledge, its experience and know-how. It is therefore necessary to optimize the place of man in the transport system in full knowledge of his capacities but also of his limits.

The accident risk analysis methods presented in Figure 1 (PHA, FSA, SEEA, FMECA, FTA, etc.) are not the only methods used in the rail transport sector. There are several other methods like HAZOP (Hazard and operability study) or What If / checklist studies and Bow-Tie Assessments. The objective of this paragraph simply aims to position the SEEA method (which is the subject of our study) in the process of safety development. For a more detailed study on the methods of analyzing the safety of industrial systems, the reader can refer to the work by A. Villemeur [2].

All of these methods, whether quantitative or qualitative, inductive or deductive, are now governed by standards and regulations with a view to meeting European rail safety and interoperability directives. For example, regulation n ° 2018/762 of the European Commission of March 8, 2018 establishes Common Safety Methods (MSC) relating to the requirements in terms of Safety Management System (SMS) in accordance with the European directive n ° 2016/798 relating to railway safety. Regarding the certification of the design, realization (manufacturing, installation, testing, acceptance) and implementation of a rail transport system, it is carried out in accordance with CENELEC EN 50126 standards (for railway systems), EN 50128 (for software of railway control and protection systems), EN 50129 (for signaling, telecommunications and processing products and systems) and EN 50657 (for on-board software of railway rolling stock).

Software must be analyzable, testable, verifiable and maintainable. Software for rail transport must have high levels of reliability, safety and integrity. The evaluation of software consists in ensuring that the behavior of the software conforms to the needs of the user. These needs may be, for example, compliance with certain safety requirements and constraints. Thus, the task of evaluating critical software is to gain complete confidence in the behavior of the software. This trust is established if it can be demonstrated that the software is safe. Remember that most of the time, for complex software, this demonstration is laborious. Consequently, the development of safety software is generally subject to compliance with certain standards.

The standard IEC / EN 61508 (entitled: “Functional safety of electrical/electronic/programmable electronic safety-related systems”) aims to develop safety applications based on electrical and electronic systems. It is generally used for the

development of "critical" software in the automation sectors as well as for industrial process control installations. Many sub-standards have been developed from IEC / EN 61508: EN 50128 and EN 50129 for the rail sector, EN 61513 for nuclear, ISO 26262 for the functional safety of road vehicles, etc.

European standard EN 50128 (entitled: “Railway applications - Communication, signaling and processing systems - Software for railway control and protection systems”) specifies the procedures and technical requirements applicable to the development of programmable electronic systems used in control and monitoring applications railway protection. The standard requires that all systems with safety implications and containing software be assigned a Software Integrity Level (SIL: Security Integration Level). The integrity of software is distributed over five SIL levels, ranging from SIL 0 to SIL 4. The software security standard EN 50128 comes from the European Committee for Electro technical Standardization (CENELEC). The published international version of the CENELEC EN 50128 standard is IEC 62279. The content of the two publications is identical.

Annex D of European Standard EN 50128 suggests 71 methods and techniques for developing and assessing the security of critical software used in the rail transport sector. The SEEA (Software Error Effect Analysis) method is part of this list of recommended methods.
According to standard EN 50128, the objective of the SEEA method consists not only in identifying the software components and their criticality, but also in proposing a means for detecting software errors and consequently reinforcing the robustness of the software.

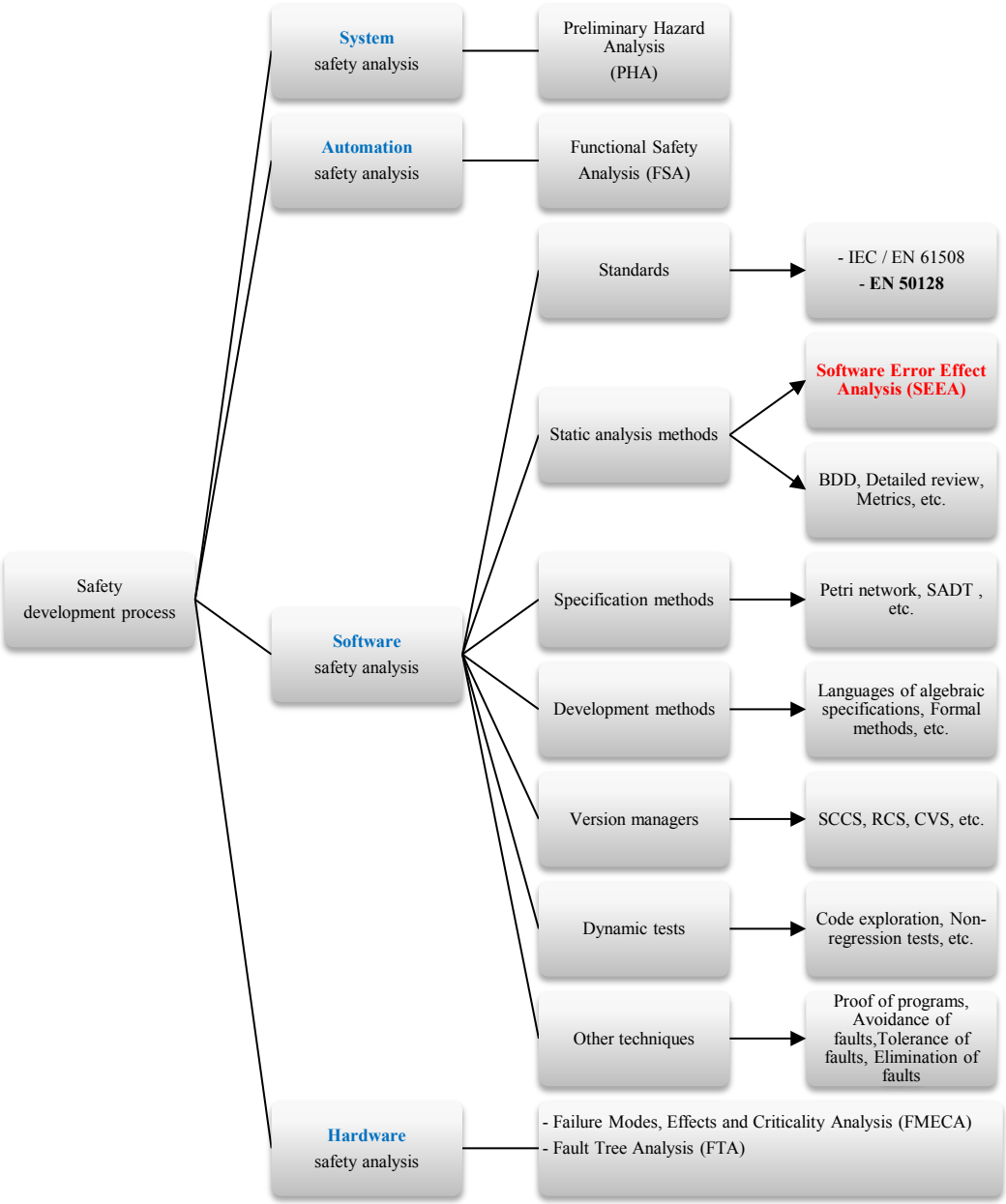

Fig. 1. Positioning of the SEEA method in the process of developing safety

3 Software Error Effect Analysis (SEEA)

The study proposed within the framework of this article concerns the SEEA method and endeavors to develop a new approach to analysis and evaluation of the safety of critical software, based on machine learning and more precisely on the Case-Based Reasoning (CBR).

It is currently impossible to conclusively demonstrate that software is free of errors. In France and in the railway sector, coded single-processor technology is used to ensure the safety of software execution. However, this technique does not provide protection against software design errors, code conformance errors, not coded safety software errors, and coded processor implementation errors. SEEA can, for its part, support, among other things, the analysis of these errors SEEA is a safety analysis approach whose purpose is to determine the nature and severity of the consequences of software failures SEEA also guides software validation and maintenance activities by identifying the most critical modules for safety. Indeed, SEEA makes it possible to estimate the level of effort of validation to be carried out on the various elements of the software and in particular, to guide the readings of code and to better target the tests. This analysis is performed by considering software error assumptions and examining the consequences of these errors on the other modules as well as any system-related failures SEEA finally proposes measures to detect errors and improve the robustness of the software. This method, which is derived from the FMECA (hardware safety analysis) method, seeks to identify in particular design and programming errors in order to analyze their internal and external effects. A SEEA is carried out according to the following three stages [3]:

- Preliminary analysis: This first step of the process consists in listing the elements of the software for which there will be a SEEA to perform, to define the levels of deepening of the analysis and finally to assign one of these levels to each element.
- Procedure-by-procedure analysis: The purpose of this analysis is to produce the SEEA files. The SEEA file contains, for each module studied, the SEEA sheets produced on this module. The development of a SEEA form consists of filling in a table containing the following columns: the name of the module studied, the error considered, the consequences on the module, the consequences at the system level, the safety criterion not respected, the criticality of the error, the means of detecting the proposed error, the criteria not respected and finally the residual criticality.
- Synthesis of the works: This last step of the SEEA approach makes it possible to group, by module, the unsolved scenarios, the criteria not respected, the means of detection and the distribution of the errors according to the criticality of their manifestation.

All of the above findings show that SEEA is considered as an important part of a system's safety record. It is a fundamental document in the process of building and validating the safety of critical software. Nevertheless, the careful analysis of certain SEEA files of already certified or approved rail transport systems reveals some shortcomings. On the one hand, SEEA documents have extremely varied representation

formats from one manufacturer to another, and on the other hand, the process of drawing up and evaluating a SEEA dossier proves to be a particularly delicate and tedious exercise which is not supported by any formalized strategy. Indeed, the completeness and coherence of the analyzes remain essentially based on the know-how, the intelligence and the intuition of the experts of the field. These findings led us to use artificial intelligence and machine learning techniques, and in particular CBR [4].
The design and implementation of this tool requires the use of four main steps [4–7]: 1) Formalization and modeling of the knowledge involved in SEEA files, 2) Acquisition of historical data from the files of manufacturers and know-how of safety experts, 3) Development of a database and 4) Operation of this database to help certification experts to judge the completeness and consistency of SEEA of a new transport system.

4 Literature review : Case-Based Reasoning (CBR)

In the field of transport, researchers have become increasingly interested in the application of AI techniques: railway maintenance [8], traffic control [9], detection of lateral rail faults [10], detection of rail surface faults [11], railway maintenance and safety [12-13], management of railway applications [14], improvement of call reporting systems [15], implementation of a predictive approach safety [16] and Siemens for using Big Data to build the Internet for trains [17]. As part of this manuscript, our research focuses on the contribution of machine learning techniques, in particular CBR, to the safety of rail transport software. The CBR is generally interpreted as an important process for solving new problems based on finding similar solutions to the problems of the past. It is part of a behavior commonly used in solving everyday human problems. Indeed, all human reasoning is generally based on past cases lived personally. Very schematically, in the context of the CBR, a case is considered a problem with his solution as well as procedures allowing a justification of the decisions made on the way the solution was generated. Generally, the CBR involves an iterative process that revolves around the next major steps: The establishment of indexes (or indexing), Search for similar cases, Reusing cases, Revision and Learning.
CBR is attracting more and more attention from researchers and experts in the transport sector. Our literature search covered three transport sectors: Air, road and rail. In the field of air transport we can cite, for example, the prediction of accidents and incidents [18]. In the road transport sector, the application of CBR is numerous: Transport planning [19], management of traffic flows [20-21], control of urban intersections to avoid road congestion [22], the analysis of road collisions [23], the improvement of traffic in urban intersections by developing new signaling plans [24], the control of traffic flow at intersections (traffic control systems (TCS)) [25], the diagnosis of the driver's stress level [26], or the modeling of the risk of driver fatigue [27]. Finally, in the rail transport sector, studies include the diagnosis of locomotive failures [28], the recovery of incident reports [29], the prevention of rail operations incidents [30], the command of railway rescue (Emergency Relief Command) [31], analysis of safety risks related to the operation of the metro [32], automatic train con-

duction to reduce travel time and save fuel consumption [33] and finally the diagnosis of failures of the rail switching system [34-35].

As part of our study on the safety analysis of critical software, the objective of improving the quality of accident risk analyzes, guided us towards the development of a tool based on the CBR allowing us to suggest potential accidents and / or the most appropriate protective or preventive measures to protect against a particular risk. However, the artificial intelligence approaches cannot provide satisfactory answers to our research objectives. Indeed, despite the undeniable interest of these approaches, to our knowledge, to date there are no applications of artificial intelligence to improve the safety of critical software used in the rail transport sector and in particular tools to improve the SEEA method. Specifically, the bibliographic study carried out on machine learning and in particular on CBR shows the absence of work on the use of CBR in the analysis and evaluation of the safety of critical software used in the rail transport sector. To date and to our knowledge, this is the first work in this area, which is one of the original features of our study presented in the next paragraph.

5 Method: Approach used for CBR-based critical software safety assessment

In order to show the interest of machine learning and more precisely CBR in the field of the safety of railway transport, we have developed a tool called "SAUTREL". This tool helps safety experts in their SEEA document analysis and assessment tasks. The design and implementation of this tool required the following three major phases:

- Acquisition of data involved in SEEA. This analysis and abstraction stage led to the development of a conceptual model based on eight descriptive parameters: system, subsystem, module, envisaged error, safety criterion, feared risk, severity of damage and finally the error detection means.
- Based on the study of two already certified rail transport systems (MAGGALY and TGV Nord), we have built up a learning example base which brings together, to date, 250 historical cases relating to SEEAs.
- Design and implementation of a tool "SAUTREL" to aid in the evaluation of SEEA [36-37]. The CBR process was implemented using Recall software from Isoft.

These three major phases of development of the tool to aid the assessment of the safety level of critical software are detailed in Figure 2 in nine steps presented in the following paragraphs. As shown in Figure 2, in front of each step of the proposed methodology, we presented the result obtained. For example, Step 1 on Knowledge Acquisition and Modeling allowed the development of a generic SEEA representation model. Step 2 on the definition of the description language of the SEEA learning examples led to the elaboration of the descriptive parameters (or characteristics) of the SEEAs. Step 3 on the development of the database the SEEA made it possible to compile all the source cases.

6 Results: Example of the mock-up use

The following paragraphs show, through an example, the use of the developed tool which involves the following nine steps (Figure 2):

1. Acquisition and modeling of knowledge,
2. Definition of the description language of the SEEA examples,
3. Development the SEEA case base,
4. Parameterization and Calibrating of the CBR process,
5. Entering the new SEEA target case for evaluation,
6. Indexing of the SEEA case base,
7. Extraction of similar SEEA cases,
8. Adaptation of extracted cases (source cases),
9. Updating the SEEA base.

6.1 Acquisition and modeling of knowledge

This paragraph presents the results of the phase of formalization and acquisition of the knowledge necessary for the development of a historical case base (experience feedback) in order to capitalize and perpetuate the knowledge related to the SEEA. The first step of the study is devoted to the research and identification of descriptors and characteristic parameters to represent and formalize the SEEA. After a second step of data collection necessary to list the possible values taken by each parameter (or descriptor), the third step proposes, a formalism of representation of SEEA documents. Finally, on the base of this formalism, which constitutes the basic language of SEEA representation, the fourth stage of the study focuses on building the case base that currently comprises 224 cases, each of which represents a particular situation that is contrary to safety (Problem) and one or more preventive measures or corrective measures to guard against, avoid, reduce, or permanently eliminate the potential risk envisaged (Solution). To leverage knowledge of SEEA (or historical cases), it is necessary to adopt a model (or formalism) that is generic enough to cover as much as possible SEEA documents (or files) from several more or less different transport systems. To build this model and in order to show the feasibility of the study, we examined the SEEA relating only to two rail transport systems already certified and put into circulation in France: the automated system “MAGGALY” and the system TVM (track-to-train transmission) of the “LGV-Nord”. It is important to emphasize that each SEEA file is specific to a particular system and therefore it is necessary to perform sufficient analysis and abstraction work to cover the majority of systems. Indeed, this analysis presents some difficulties, since from one manufacturer to another, or even from one system to another, the formalism, the terminology or the level of deepening of the analysis implemented are different. At the end of this review, we finally proposed a first SEEA representation model that relies heavily on the manufacturers’ practices and our experience in the field of railway safety. This formalism is based on eight characteristic parameters: Studied system, subsystem studied, module studied, error envisaged (family, class, type), safety criterion not respected by the error, dreaded event, type and gravity of the damage, barrier and means for detecting

the error. This model proposes a methodological framework for preparing SEEA files and thus contributes to ensuring the quality of future analyzes. An excerpt from this formalism is presented in Figure 3. On the basis of this representation model of the SEEA forms, we have created a library of 224 typical cases.

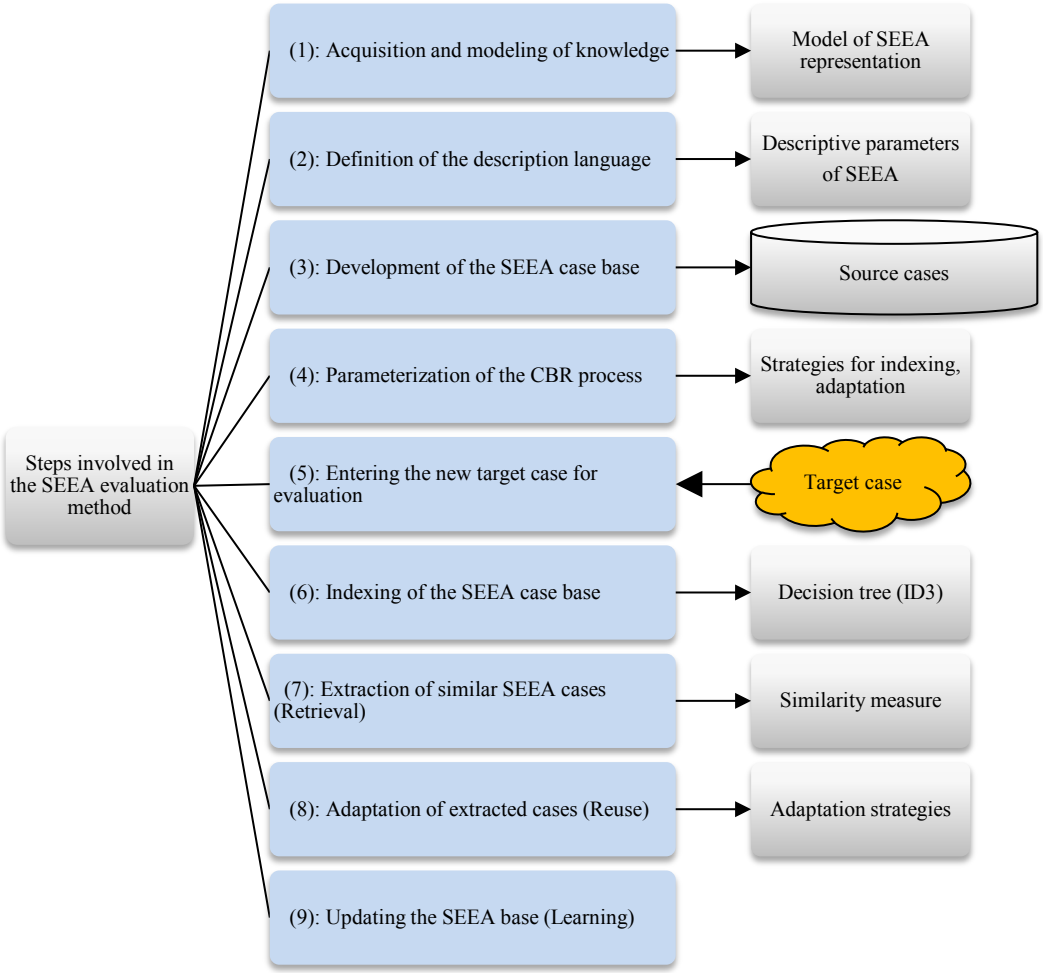

Fig. 2. Approach for acquisition, modeling, capitalization and assessment of SEEA

6.2 Definition of the description language of the SEEA examples

This step allows you to enter the description language of an SEEA based on the eight descriptors listed above (Figure 3). A descriptor is a couple (attribute, value). All attributes are symbolic. Three types of descriptors could be distinguished: Enumerated descriptors, multi-valued descriptors and unknown descriptors.

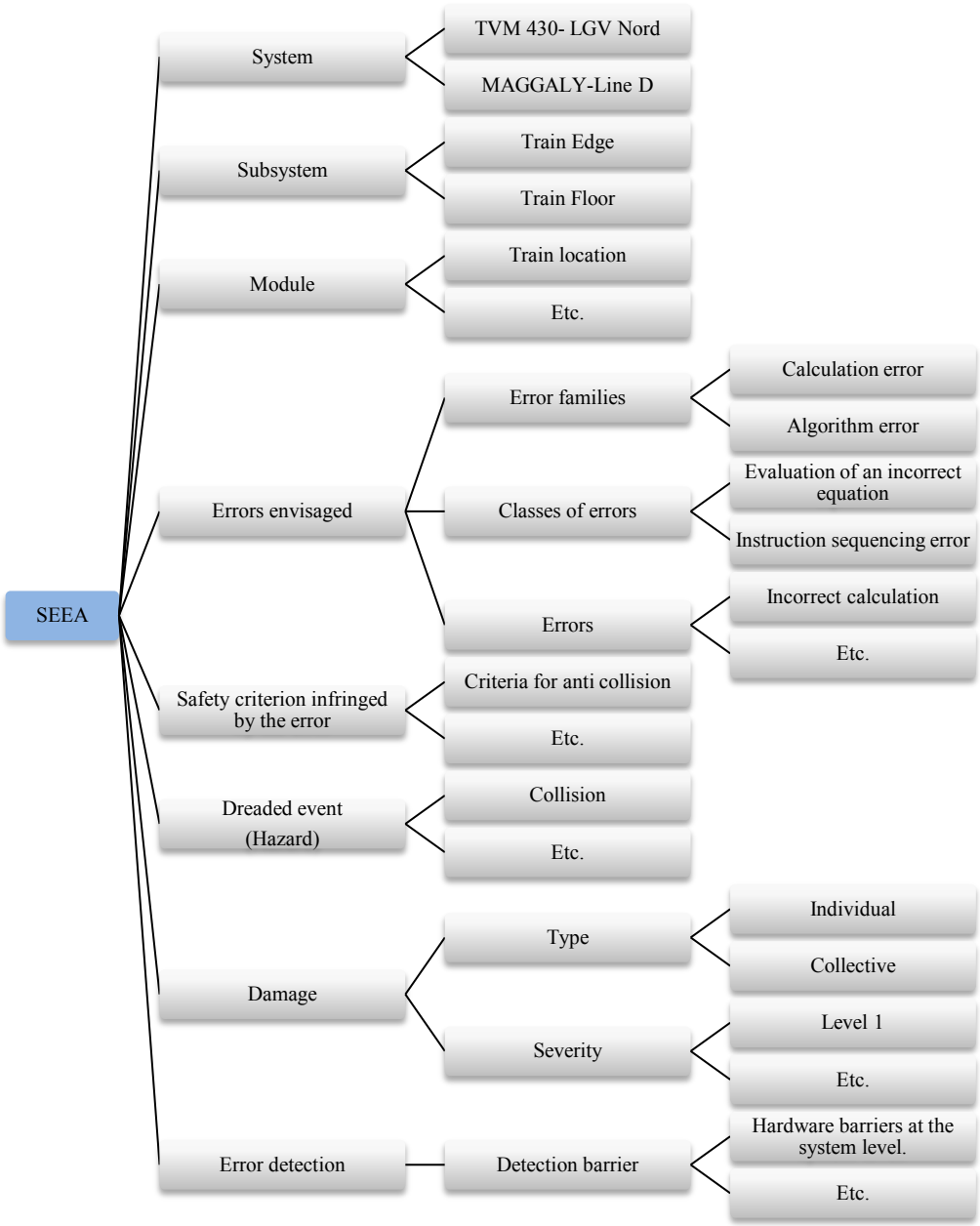

Fig. 3. Extract from the formalism elaborated for the representation of records SEEA

6.3 Developing the SEEA case base

During this step, it's about creating cases by assigning a value to each attribute of the description language. This case base may subsequently be modified or consulted. The acquisition of the target case is done by entering the value or values of the different attributes. During this case base construction step, the concept descriptor "dreaded event" is left unknown because it represents the solution we are looking for in the case base.

6.4 Parameterization of the CBR process

During this step, the user must set different parameters to configure the CBR process. These choices concern both the descriptor that will represent the solution of the problem and the strategies of indexing, matching or adaptation. During this step, the user must set the following parameters:

- The descriptor "concept": The user must choose from all the descriptors the one that will represent the solution of the problem. In our example, the descriptor "concept" is the descriptor "dreaded event". The problem, meanwhile, will be characterized by all the other descriptors.
- Indexing strategies: In general, we use indexing rules that make it possible, on the one hand, to organize the case memory and, on the other hand, to express the relevant characteristics of the entries (the target cases) in terms of indexes. The process of extracting or choosing the source case strongly depends on the quality of the organization of the case memory. The memory organization mechanisms use several indexing techniques such as "Memory in bulk" or "Hierarchical memory". In the context of the "Memory in bulk", we use a sequential search algorithm which consists, for all stored cases, of comparing the target case with the extracted case. It returns the most similar cases. The exploration is systematic and it is very easy to add a case but extracting one is very expensive because the memory must be covered entirely. In the context of the second indexing mechanism based on a "hierarchical memory", cases are accessed through a tree or an indexing graph. Each node of the tree corresponds to a logical partition of the case base. Finding the most similar set of cases returns to the level of each node, finding the best son of the tree. This method is effective in search time, but it is more difficult to add a case (it must be inserted into the tree in the right place). In our sample application, the tool offers several strategies for prioritizing memory. The user can set this hierarchy by sorting the descriptors or trimming the hierarchy. In our example, we construct the hierarchy by taking into account all the descriptors and by imposing the descriptors "studied system" and "studied subsystem", in this order, as first and second level of the decision tree. Then, the choice between the remaining descriptors for the next levels will be done by a decision tree classification algorithm: Quinlan ID3 algorithm [53].
- Matching strategies to search for similar cases: Given a new problem to be solved (target case), it is, from a known case base (source cases), to find the most similar case (s) and relevant to solve the new problem. In this step, we gen-

erally use matching rules or similarity measures such as the “connective model” which imposes on each of the characteristics of the target case to be sufficiently close to all those of the source case or else “the disjunctive model” which evaluates the source case on its particularity closest to that of the target case. In this case, a source case will be considered acceptable if it is very close to the target case on at least one relevant characteristic, regardless of the value of the others. Most CBR systems evaluate the similarity of two cases by accounting for their common characteristics: This is the Euclidean distance. In our sample application, the user can intervene in several ways in calculating the similarity between two attributes. It can possibly specify the descriptors which will not have to be taken into account during the computation. It can also give a weight vector to indicate the relative importance of a descriptor over others. In our example, we chose to extract only the 10 most similar cases, and to give a weight equivalent to all the descriptors.

– Adaptation strategies for reuse of similar cases: Generally, there are two possibilities: 1) if the case found in the database (source case) is identical to the new problem to be solved (target case), then the solution of the problem is immediate; 2) if, on the other hand, the case found presents a certain similitude (or analogy) with the new case, then an adaptation procedure is necessary whose objective is to adapt the solution found to the need of the new situation (target case). Thus, in the first hypothesis, we apply directly the solution found and in the second hypothesis, we must find a suitable technique to adapt the recovered solution and include it in the new problem. In our sample application, to date, the tool does not offer a real adaptation method, but allows the user to program his own methods with daemons. Currently, this adaptation can be done either implicitly by the safety domain expert, by comparing cases similar to the target case, or by the voting technique. In this second case, the value of the attribute to be adapted is calculated on all the similar cases by a vote weighted by the percentage of similarity of each case. For example, if a case C has 3 descriptors of which 2 are 100% similar to the target case and the third descriptor has no similarity (0%), then case C will be similar with the target case at 66%. If all the descriptors are of equal weight: (100 × Descriptor weight 1 + 100 × Descriptor weight 2 + 0 × Descriptor weight 3)/3 = 66.

6.5 Entering the new SEEA target case for evaluation

The acquisition of the target case is done by entering the value or values of the different attributes. We will leave the concept descriptor “dreaded event” unknown because it represents the solution we are looking for in the source case base.

6.6 Indexing of the SEEA case base

After developing the SEEA case representation mode, i.e. the description of the problem and the solution in the form of descriptors (attribute/value), it is then necessary to build a model for organizing and indexing the memory. This model is essential in the

search for similar cases and must have certain qualities. Knowing that the research phase of similar cases must keep a constant complexity as the case base is filled; it is wise to consider a solution to quickly find similar cases. To apprehend this problem, we use the indexing method where each node of the tree corresponds to a question on one of the indexes and the threads of the tree correspond to the different answers. An index represents the elements discriminating the cases and has two fields: its name and its value. To ensure a minimum of efficiency, the tree, which is dynamically built, must ask the questions in the right order and be as shallow as possible. The best way to build it is to use the decision tree method. Decision tree consists of nodes corresponding to the attributes of the selected objects and branches characterizing the alternative values of these attributes. The leaves of the tree represent the sets of objects of the same class of objects. The construction of decision trees is a top down generalization approach. The ID3 of QUINLAN algorithm [38] is a typical case of a downward approach. ID3 uses a heuristic search strategy, according to the gradient method, by optimizing a numerical criterion called gain of information which is based on the entropy of SHANNON developed in the early 1940s by Claude Shannon [39].
From:

- A set of exclusive classes {C1, C2, ... Ck};
- A set of examples {E1, E2, ... En} represented in the form of pairs (attribute/value) and partitioned in classes Ci;

ID3 produces a decision tree that allows to recognize (or classify) all the examples E_i. This tree can then be used to generate classification rules.
QUINLAN's method consists in successively testing each attribute to know which one to use first in order to optimize the gain of information. That is, the attribute that best distinguishes between examples of different classes. This principle has been applied in many cases and has contributed to the development of several expert systems, essentially dedicated to diagnosis. Subsequently, work was devoted to improving the principle of construction of the decision tree and in particular reducing the size of the tree, improving the selection strategy (which is based in ID3 only on the attribute) by proposing a selection based on both the pair (attribute/value) or the improvement of the representation mode of the examples, by using a representation based on diagrams (frames). Used in a variety of fields such as data mining, business intelligence, medicine, safety, etc., the decision tree is a decision support tool that represents a set of choices in the form of graphical data (tree). In our case of application to SEEA, we use the classification algorithm ID3. During this indexing or prioritization step, the user selects the case base to index, and then starts the construction of the hierarchy. In our example (Figure 4), the first two levels of the hierarchy are constructed from the descriptors "studied system" and "studied subsystem". Here, the third level deals with the descriptor "Severity of the damage".

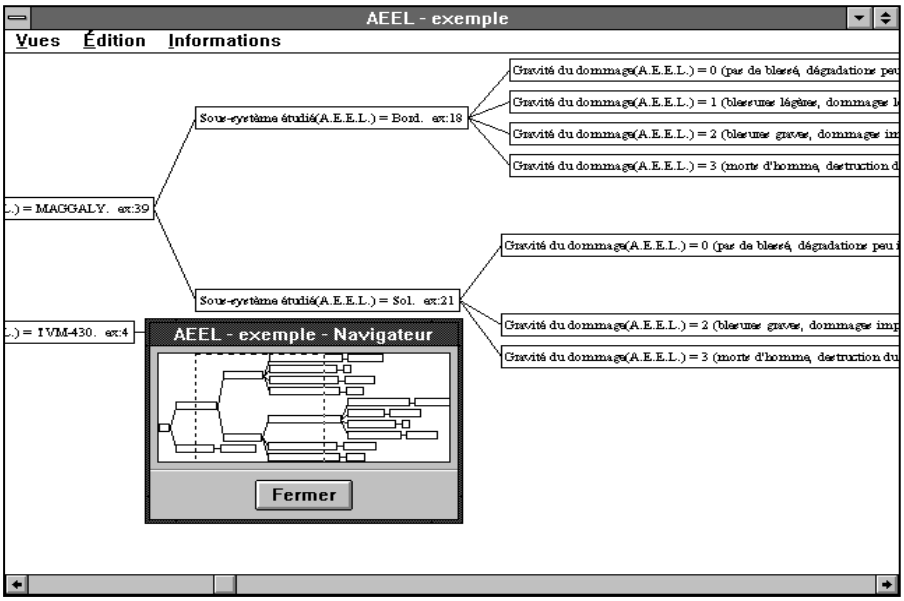

Fig. 4. Example of the instances base hierarchy

6.7 Extraction of similar SEEA cases

~~The Before~~ Searching for similar cases, if some information is missing (for example, a value of an attribute not specified), it is possible to complete the knowledge acquisition phase by querying the domain expert. There are some learning tools to try to determine and correct this data. In our case of application, during the phase of acquisition and collection of SEEA data, particular attention was paid to this problem of noisy or inconsistent data. The search for SEEA cases similar to the target case, is broken down into two filtering and selection stages that use static and dynamic indexes. There are different ways to determine the characteristics of indexes: All characteristics, some characteristics, the most discriminating characteristics, etc. In our application we adopted a similarity search based on the set of characteristics. To find similar SEEA cases from the case database archived in memory (source cases or reference cases), several techniques can be used, such as the "Nearest Neighbor" algorithm whose objective is to measure the similarity between the problem (target case) and potential source cases. The comparison method is based on the indexes. Thus, from the similarity on each index, the algorithm generates the global similarity sought. Let's remember that the search for nearest neighbors, or k nearest neighbors commonly used in machine learning, consists of starting from a set of other points to find the nearest K (similar) points. Generally, to optimize this method, we use heuristics and selection strategies to quickly find the most useful cases to solve the problem. The cases that share the most important characteristics, the easiest cases to adapt or the most used cases are examples of heuristics. In our application example, from the historical case base (source cases), it is a question of finding the SEEA cases most simi-

lar to the SEEA cases to be evaluated (target case) and who share the most important characteristics. The screen shown in Figure 5 shows, for our example, the result search for similar cases. The target case is recalled in the right column, the left column proposes the first 10 most similar cases and the middle column shows one of the similar cases (here case 33).

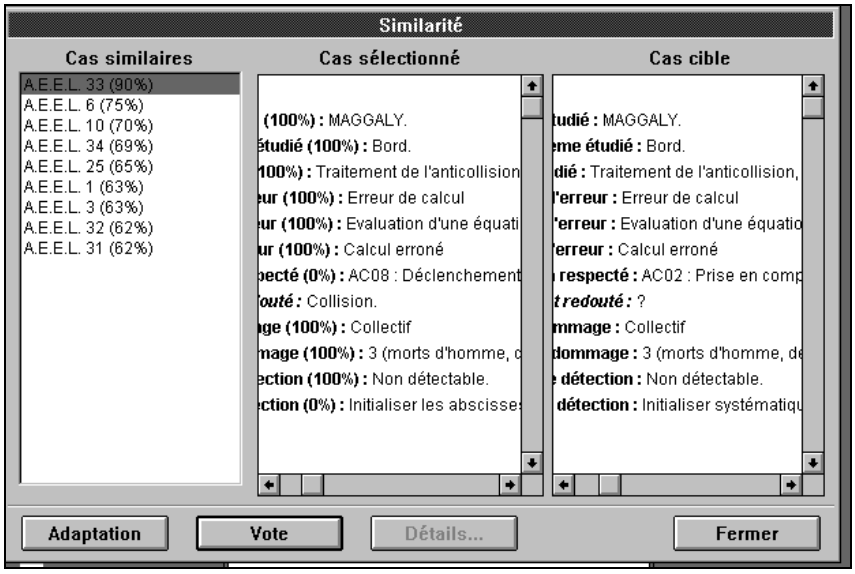

Fig. 5. Visualization of similar cases extracted from the case base

6.8 Adaptation of extracted cases (source cases)

Suppose we found a similar case, so we reuse directly the solution he proposes to solve the problem (case target). In practice, it is often rare that we find a case identical to the problem, so it is necessary to adapt pre-existing solutions. Adaptation therefore consists of building a new solution from the target case and similar cases found. It is then necessary not only to look for the difference between the cases found (source cases) and the problem, but also to find the useful information to be transferred to the new solution. Generally, one distinguishes two types of adaptation: Transformational adaptation and derivative adaptation. In the first approach, it is a question of directly reusing the solutions of the past cases. This type of transformational adaptation does not tell us how the solutions of similar cases were generated. It is the role of derived adaptation that allows, for each case stored in the database, to explain the reasoning process leading to the solutions. In this case, the derivative adaptation consists in applying the same reasoning to the new problem by choosing the paths taken by the old solutions selected and thus avoiding any unsuccessful paths. In our application case, the “ReCall” tool used to demonstrate the feasibility of the proposed approach does not yet propose relevant adaptation strategies. To date, the adaptation phase is still assigned to the user and in particular to the safety expert. With the screen presented in Figure 6, the user can consult the value taken by the concept attribute

"dreaded event" in each similar case and choose himself the value to give to the "concept" attribute for the target case. The user can also use the voting technique. In our example, the tool proposes a single value for the attribute "dreaded event": Train collision. Thus, the domain expert can adapt the most similar case (proposed by the tool) by assigning the "Feared Event" concept the value "Collision" as a solution to the problem. Since the "ReCall" tool does not propose adaptation strategies, the adaptation phase is limited in our example to indicate the class of potential solution. The solution sought is therefore focused simply on the value of the concept "feared event" proposed by the tool: "collision". Nevertheless, this knowledge is necessary to stimulate and assist the expert in his task of safety assessment. Indeed, faced with a new problem (scenarios of accident/potential incident) described by a set of characteristic descriptors, it is interesting to know the possible feared event or events (collision, derailment, electrocution, fall).

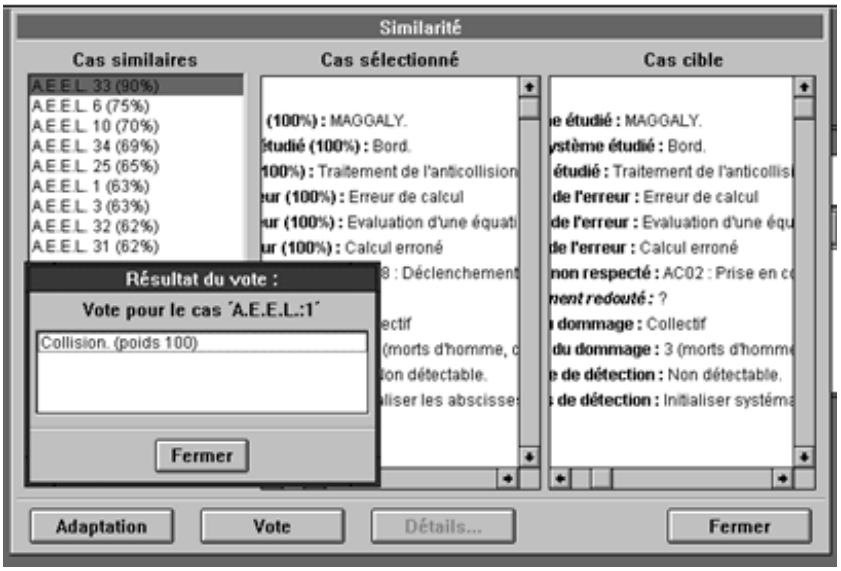

Fig. 6. Example of the reference cases consultation and the vote technique use

6.9 Updating the SEEA base

This last step of updating knowledge is to perform the automatic learning by adding the appropriate target case in the SEEA historical case base. In the "ReCall" software, this learning is not incremental since the new case will be integrated into the hierarchy without it being reconstructed. It is up to the user to take the initiative to revive the indexing of the case base. Therefore, during this phase of the CBR cycle, it is wiser that the new case with its new solution is validated by the domain expert before being added to the case base (source cases). In addition, it is interesting at the end of this learning phase to test the system by relying on the same problem that it has just treated to ensure that the system behaves as expected. Finally, it is essential to determine how to index this new case in the database without questioning the historical

knowledge learned in previous phases and thus avoid new problems of inconsistency, redundancy, etc. In particular, the focus must be on this problem of incrementality. Should we adopt a monotonous incremental learning approach (accumulation of knowledge without questioning knowledge previously learned) or non-monotonous (examination of knowledge learned with each addition of new knowledge)? This is a problem that remains crucial in almost all machine learning systems. As part of our prototype of feasibility, this work is not yet completed.

7 Discussion and Conclusion

~~Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either.~~

In order to rationalize and reinforce conventional approaches to safety analysis and assessment, we have agreed to use artificial intelligence and machine learning techniques and in particular case-based reasoning (CBR). The main objective consists, from a set of data in the form of accident scenarios or incidents experienced on rail transport systems (experience feedback), to exploit by automatic learning this mass of data in to stimulate the imagination of safety experts and assist them in their difficult task of analyzing and evaluating the safety of new critical software. This historical data concerns SEEA. The implementation of this railway safety assessment approach required not only the use of machine learning but also knowledge acquisition methods to collect, structure and formalize the knowledge involved in SEEA. The knowledge acquisition phase ultimately culminated in the implementation of a conceptual SEEA representation model that provides a methodological framework for safety experts. Based on this model, we acquired 224 cases of SEEA (historical basis for learning). This learning base is based on experience feedback from two rail transport systems put into service in France. The first Maggaly system in Lyon is fully automated and the second system relates to a High Speed Line (TGV-Nord). When it comes to machine learning, our work is part of supervised learning. Indeed, the presence of the safety expert is essential to ensure effective and relevant learning. The domain expert is not only able to control, validate, adapt and complete the knowledge learned by the system, but also to adjust certain learning parameters. To demonstrate the feasibility of the proposed approach, we used a case-based reasoning generator named “ReCall” from ISOFT. Despite the undeniable interest of this ReCall tool, several shortcomings have been noted in particular for methods for calculating similarity, coping strategies and processing missing values (noisy data). However, this contribution made it possible to demonstrate the feasibility of a new approach to modeling, capitalization and evaluation of the SEEA method, based on the use of machine learning techniques. This approach of evaluating critical software, used in rail safety, is also based on the joint and complementary use of machine learning and knowledge acquisition techniques to reinforce and systematize the phase of acquisition and transfer of knowledge in the field of railway safety. The originality of the tool developed lies not only in its ability to model, capitalize, sustain and disseminate SEEA expertise, but to the best of our knowledge, it represents the first research on the application of CBR to SEEA. In

fact, in the field of rail transport, there are currently no software tools for assisting SEEAs based on machine learning techniques and in particular based on CBR. Currently, project is at the mock-up stage. Initial validation has demonstrated the interest of the suggested approaches, but improvements and extensions are required before they could be used in an industrial environment or adapted to other areas where the problem of investigating safety arises. These improvements include the improvement of the adaptation strategies of the solutions proposed by the system, the enrichment of the SEEA case base to cover the whole problem and finally, it is necessary to construct an integrated version of a prototype in order to finalize the results of demonstration model.

Although new technologies have progressively reduced rail operating accidents, many rail accidents have been caused by degraded human performance and human error, and the tasks of drivers, signalers and controllers have remained essentially the same. Railway safety should be generally maintained and constantly improved taking into account not only technical and scientific progress, but also the impact of human error on the transport system. Indeed, with the increasing complexity of industrial systems and especially guided or automated rail transport systems, considerable evolutions have taken place in the way of thinking and understanding the role and place of man in the safety of human-machine systems. In this context, human factors play an important role in safety analyzes and especially after the occurrence of accidents (feedback of experience) that sometimes lead to human losses and the destruction of the environment and system equipment.
In order to provide an element of response to these problems linked to human and organizational factors and complete this work, we have developed a new methodology for analysis, classification and evaluation of human errors involved in the safety of guided rail transport [40].

References

1. Darricau, M., Hadj-Mabrouk, H., Ganascia J-G. : Méthodes contribuant à l'évaluation des logiciels de sécurité - Étude bibliographique, Rapport de convention INRETS/LAFORIA-LIP6, n° ESTAS A/96-62, 44p, (1996)
2. Villemeur, A.: Sûreté de fonctionnement des systèmes industriels, Paris, Eyrolles, coll. "Collection de la direction des études et recherches d'Électricité de France", ISSN 0399-4198, 744p, (1988)
3. Thireau, P.: Méthodologie d'Analyse des Effets des Erreurs du Logiciel (AEEL) appliquée à l'étude d'un logiciel de haute sécurité. 5° colloque international de fiabilité et de maintenabilité, Biarritz, France (1986)
4. Hadj-Mabrouk, H. : Contribution du raisonnement à partir de cas à l'analyse des effets des erreurs du logiciel. Application à la sécurité des transports ferroviaires, Ouvrage collectif, Raisonnement à partir de cas, Vol. 2, chapitre 4, Éditions Hermes/Lavoisier, pp. 123–148 (2007)
5. Hadj-Mabrouk, H.: Machine learning from experience feedback on accidents in transport. 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 246–251 (2017)

6. Hadj-Mabrouk, H.: Contribution of learning Charade system of rules for the prevention of rail accidents. Intell Decis Technol, Vol. 11, pp. 477–485 (2017)
7. Hadj-Mabrouk, H.: A Hybrid Approach for the Prevention of Railway Accidents Based on Artificial Intelligence, In: Vasant P, Zelinka I, Weber GW (eds.), International Conference on Intelligent Computing & Optimization, pp. 383–394 (2018)
8. Jamal, S., Goyal, S., Grover, A., et al.: Machine Learning: What, Why, and How? In: Shanker A (Eds.), Bioinformatics: Sequences, Structures, Phylogeny, Springer, Singapore, 359–374 (2018)
9. Bergmeir, C., Sáinz, G., Bertrand, C-M., et al. : A Study on the Use of Machine Learning Methods for Incidence Prediction in High-Speed Train Tracks, In: Ali M, Bosse T, Hindriks KV, Hoogendoorn M, Jonker CM, Treur J (eds.), Recent Trends in Applied Artificial Intelligence, IEA/AIE 2013, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 7906: 674–683 (2013)
10. Fay, A.: A fuzzy knowledge-based system for railway traffic control. Eng Appl Artif Intel 13: 719–729 (2000)
11. Santur, Y., Karaköse, M., Akin, E.: A new rail inspection method based on deep learning using laser cameras. International Artificial Intelligence and Data Processing Symposium (IDAP), 16–17 (2017)
12. Faghih-Roohi, S., Hajizadeh, S., Núñez, A., et al.: Deep convolutional neural networks for detection of rail surface defects. International Joint Conference on Neural Networks (IJCNN), 24–29 (2016)
13. Ghofrania, F., He, Q., Goverde, R., et al.: Recent applications of big data analytics in railway transportation systems: A survey. Transport Res C-Emer 90: 226–246 (2018)
14. Thaduri, A., Galar, D., Kumar, U.: Railway assets: A potential domain for big data analytics. Procedia Comput Sci 53: 457–467 (2015)
15. Attoh-Okine, N.: Big data challenges in railway engineering. IEEE International Conference on Big Data (Big Data), 27–30 Oct. 2014, Washington, DC, USA. (2014)
16. Hughes, P.: Making the railway safer with big data. Available from: http://www.railtechnologymagazine.com/Comment/making-the-railway-safer-with-big-data. (2018)
17. Hayward, V.: Big data & the Digital Railway. Available from: https://on-trac.co.uk/big-data-digital-railway/ (2018)
18. Marr, B.: How Siemens Is Using Big Data And IoT To Build The Internet Of Trains. Available from: https://www.forbes.com/sites/bernardmarr/2017/05/30/how-siemens-is-using-big-data-and-iot-to-build-the-internet-of-trains/#2b7a4b6e72b8. (2017)
19. Zubair, M., Khan, M-J.: Awais M Prediction and analysis of air incidents and accidents using case-based reasoning. Third Global Congress on Intelligent Systems, 6–8 Nov., Wuhan, China. (2012)
20. Khattak, A., Kanafani, A.: Case-based reasoning: A planning tool for intelligent transportation systems. Transport Res C-Emer 4: 267–288 (1996)
21. Sadeka, A., Smith, B., Demetsky, M.: A prototype case-based reasoning system for real-time freeway traffic routing. Transport Res C-Emer 9: 353–380. (2001)
22. Sadek, A., Demetsky, M., Smith, B.: Case-Based Reasoning for Real-Time Traffic Flow Management. Comput-Aided Civ Inf. (2002)
23. Zhenlong, L., Xiaohua, Z.: A case-based reasoning approach to urban intersection control. 7th World Congress on Intelligent Control and Automation, 25–27 June, Chongqing, China. (2008)
24. Li, K., Waters, N-M.: Transportation Networks, Case-Based Reasoning and Traffic Collision Analysis: A Methodology for the 21st Century, In: Reggiani A, Schintler LA (eds.),

Methods and Models in Transport and Telecommunications, Advances in Spatial Science. Springer, Berlin, Heidelberg, 63–92 (2005)
25. Kofod-Petersen, A., Andersen, O-J., Aamodt, A.: Case-Based Reasoning for Improving Traffic Flow in Urban Intersections, In: Lamontagne L, Plaza E (eds.), Case-Based Reasoning Research and Development, ICCBR 2014, Lecture Notes in Computer Science, Springer, Cham, 8765: 215–229 (2014)
26. Louati, A., Elkosantini, S., Darmoul, S., et al.: A case-based reasoning system to control traffic at signalized intersections. IFAC-Papers On Line 49: 149–154 (2016)
27. Begum, S., Ahmed, M-U., Funk, P., et al.: Mental state monitoring system for the professional drivers based on Heart Rate Variability analysis and Case-Based Reasoning. Federated Conference on Computer Science and Information Systems (FedCSIS), 9–12 Sept., Wroclaw, Poland. (2012)
28. Zhong, Q., Zhang, G.: A Case-Based Approach for Modelling the Risk of Driver Fatigue, In: Shi Z, Goertzel B, Feng J (eds.), Intelligence Science I. ICIS 2017. IFIP Advances in Information and Communication Technology, Springer, Cham, 510: 45–56 (2017)
29. Varma, A., Roddy, N.: ICARUS: Design and deployment of a case-based reasoning system for locomotive diagnostics. Eng Appl Artif Intel 12: 681–690 (1999)
30. Johnson, C.: Using case-based reasoning to support the indexing and retrieval of incident reports. Proceeding of European Safety and Reliability Conference (ESREL 2000): Foresight and Precaution, Balkema, Rotterdam, the Netherlands, 1387–1394. (2000)
31. Cui, Y., Tang, Z., Dai, H.: Case-based reasoning and rule-based reasoning for railway incidents prevention. Proceedings of ICSSSM '05. 2005 International Conference on Services Systems and Services Management, 13–15, Chongqing, China. (2005)
32. Li, X., Yu, K.: The research of intelligent Decision Support system based on Case-based Reasoning in the Railway Rescue Command System. International Conference on Intelligent Control and Information Processing, 13–15 Aug., Dalian, China. (2010)
33. Lu, Y., Li, Q., Xiao, W.: Case-based reasoning for automated safety risk analysis on subway operation: Case representation and retrieval. Safety Sci 57: 75–81. (2013)
34. De Souza, VDM. Borges, AP., Sato, DMV. et al.: Automatic knowledge learning using Case-Based Reasoning: A case study approach to automatic train conduction. International Joint Conference on Neural Networks (IJCNN), 24–29 July (2016)
35. Zhao, H., Chen, H., Dong, W., et al.: Fault diagnosis of rail turnout system based on case-based reasoning with compound distance methods. 29th Chinese Control and Decision Conference (CCDC), 28–30 May (2017)
36. Darricau, M. : Apport du raisonnement à partir de cas à l'analyse des effets des erreurs de logiciels. Application à la sécurité des logiciels critiques, Rapport de fin d'études d'ingénieur, INRETS-IFSTTAR (1995)
37. Darricau, M., Hadj-Mabrouk, H.: Applying case-based reasoning to the storing and assessment of software error-effect analysis in railway Systems. Comprail 96, 5th International Conference on Computer-Aided Design, Construction and Operation in Railway Transport Systems, Berlin, 483–492 (1996)
38. Quinlan, J-R.: Induction of Decision Trees. Mach Learn 1: 81–106. (1986)
39. Shannon, CEA: mathematical theory of communication. Bell Syst Tech J 27: 379–423. (1948)
40. Hadj-Mabrouk, H.: New approach of assessing human errors in railways. Transactions of the VSB - Technical University of Ostrava, Safety Engineering Series, 13(2): 1–17, (2018), DOI: 10.2478/tvsbses-2018-0007.