



*Research article*

## **Contribution of artificial intelligence and machine learning to the assessment of the safety of critical software used in railway transport**

**Habib Hadj-Mabrouk\***

French institute of science and technology for transport, spatial planning, development and networks, Scientific Direction, 14/20 Boulevard Newton, 77447 Marne la Vallée, France

\* **Correspondence:** Email: [habib.hadj-mabrouk@ifsttar.fr](mailto:habib.hadj-mabrouk@ifsttar.fr).

**Abstract:** As part of the process of certification and commissioning of a new guided or automated rail transport system, the domain experts and in particular the National Safety Authority are responsible for reviewing the safety of the system to ensure that the safety level of the new transport system is at least equivalent to the railway systems already in service and deemed safe. This critical task of evaluating safety essentially concerns all the safety files prepared by the manufacturer and in particular safety studies such as the Preliminary Hazard Analysis (PHA), the functional safety analysis (FSA), the analysis of failure modes, their effects and of their criticality (AFMEC) or Software Error Effect Analysis (SEEA). The study presented in this paper is part of the SEEA analysis. To respect the completeness and consistency of this safety analysis (SEEA), the experts carry out complementary analyses of safety. They are brought to imagine new scenarios of potential accidents to perfect the exhaustiveness of the safety studies. In this process, one of the difficulties then consists in finding the abnormal scenarios being able to lead to a particular potential accident. This is the fundamental point that motivated this work. To help experts in this complex process of evaluating safety studies, we agreed to use artificial intelligence techniques and in particular machine learning to systematize, streamline and strengthen conventional approaches to safety analysis and critical software certification. The approach which was adopted in order to design and implement an assistance tool for safety analysis involved the following two main activities:

- Extracting, formalizing and storing hazardous situations to produce a library of standard cases which covers the entire problem. This process entailed the use of knowledge acquisition techniques;
- Exploiting the stored historical knowledge in order to develop safety analysis know-how which can assist experts to judge the thoroughness of the manufacturer's suggested safety analysis. This second activity involves the use of machine learning techniques in particular the use of

---

case-based reasoning (CBR).

**Keywords:** artificial intelligence; machine learning; case-based reasoning; Software Error Effect Analysis (SEEA); safety of rail

---

## 1. Introduction

The process of building the safety of a rail transportation system involves three major safety analysis activities: System level analysis, software level analysis, and hardware level analysis. Our research is part of the software safety analysis and focuses more specifically on the Software Error Effect Analysis (SEEA) method. The objective of the study is the development of a KBS based on case-based reasoning (CBR) to help safety and certification experts judge the comprehensiveness and consistency of SEEA critics of a new rail transport system. In recent years KBS have achieved a notable presence within the industry and are no longer considered to be rare products of research laboratories. However, it is unusual for these systems to reach the level of performance of human experts and they frequently fail to answer the real needs of end users. This is due to the difficulty of extracting the required knowledge from one or more experts in the domain and representing this knowledge without distortion in order to produce a cognitive model of the expert. A number of research projects have described the problem of collecting and formalizing the knowledge which is handled by experts while solving a problem. Experts may find it very difficult to describe in clear terms the stages of reasoning which they go through in order to make decisions. Such a description requires experts to undertake a long process of thought which will enable them to explain the unconscious aspect of their activities. The success of a KBS project depends on this difficult and sometimes painful task. In view of the complexity of the knowledge of experts and the difficulty which they have in explaining their mental processes there is a danger that the extracted knowledge will be either incorrect, incomplete or even inconsistent. A variety of research in Artificial Intelligence (AI) is in progress in an attempt to understand this problem of the transfer of expertise. Research is currently taking place in two major independent areas:

- Knowledge acquisition, which aims to define methods for achieving a better grasp of the transfer of expertise. These methods chiefly involve software engineering and cognitive psychology;
- Machine learning, which involves the use of inductive, deductive, abductive or analogical techniques in order to provide the KBS with learning capacities.

In order to develop a tool based on Case-Based reasoning which aids in safety analysis we combined these two approaches and used them in a complementary way. This paper presents a mock-up of a tool for storing and assessing SEEA for the safety of automatic devices of terrestrial guided transport system. The purpose of our work is to exploit historical SEEA, which has already been carried out on approved safety-critical software, in order to assess SEEA of new software. The production of this mock-up involves the use of CBR. The basic principle of CBR is to deal with a new problem by remembering similar experiences which have occurred in the past. 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. We present successively knowledge-based systems, knowledge acquisition and machine learning. Emphasis is placed on the CBR, since the critical software safety Assessment Tool (SEEA) is based on the use of CBR. This same paragraph presents several examples of AI and 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. The developed methodology involves the following steps: 1) acquisition and modeling of knowledge, 2) definition of the description language of the SEEA examples, 3) development of the SEEA case base, 4) parameterization and Calibrating of the CBR process, 5) enter the new target case to evaluate, 6) indexing the case base, 7) extraction of similar cases, 8) adaptation of extracted cases and 9) updating the SEEA base. 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. The results obtained to date are presented in the last paragraph.

## **2. Main methods of railway safety analysis**

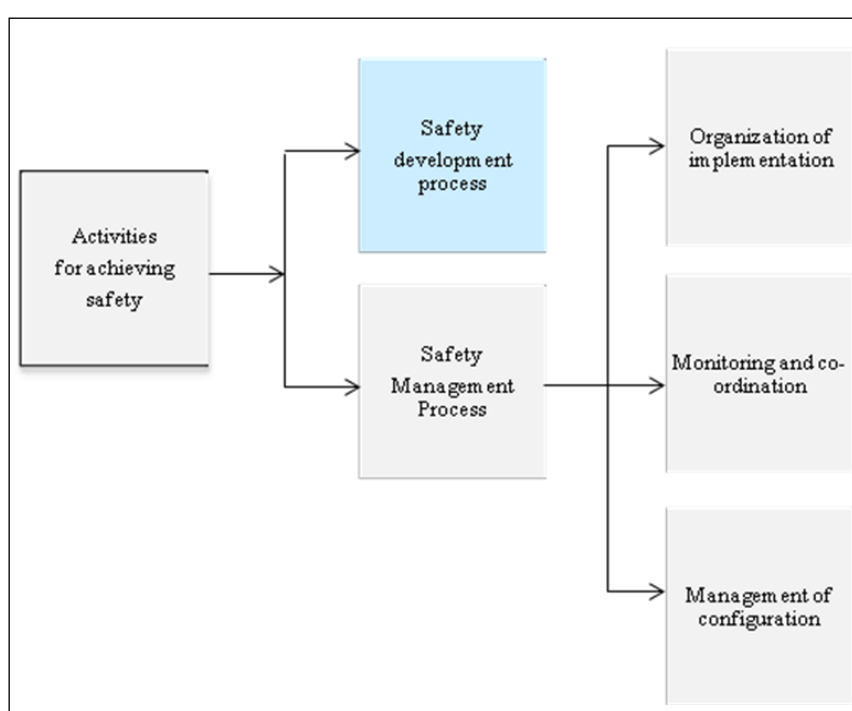
### *2.1. Notions of accident, near-accident, danger, risk and safety*

Before presenting the main safety analysis methods used in the rail transport sector, it is worth recalling some basic definitions and concepts related to safety and railway risk management. A “near-accident” or incident is an event or sequence of events causing no damage, but such that a condition that was not under control during the event or events could have led to one or more “damage”. These two notions (accident and near-accident) are generally grouped under the term “potential accident”, which is therefore an accident or a near-accident. A “danger” is then defined as a situation prior to a potential accident. The seriousness of the damage caused by an accident is measured on a qualitative scale, usually comprising four levels. For example: “minor”, “major”, “critical”, “catastrophic”. Similarly, the “likelihood of occurrence” of a potential accident is assessed by referring to a qualitative and/or quantitative scale of five or six levels of occurrence divided between “extremely unlikely” and “frequent”. The quantitative evaluation of the probabilities of occurrence is, in turn, practically impossible in the early design phases of the system and can only be obtained after further study and in particular from technical investigations after accidents or incidents (Experience feedback). In this context, the notion of “risk” is ultimately the combination of the probability of occurrence of a potential accident and the severity of the most severe damage that could be caused by this potential accident. It is usually expressed on a scale with several levels of risk: “negligible”, “unacceptable”, “tolerable”, “acceptable under certain conditions” and acceptable.

This “level of risk” can also be measured in the number of probable deaths per pre-defined time unit. There remains the notion of “safety”, which is defined by the European standard CENELEC 50129 [1] by the absence of any unacceptable level of risk, when the risk is measured on a qualitative scale.

## 2.2. Methods of railway safety analysis

In terms of railway safety, there are two major safety activities (Figure 1). The first activity is usually called the development or “construction of safety” process and the second activity focuses on managing safety (coordination, organization, etc.). The safety development process is, in turn, hierarchically structured into four safety analysis activities: 1) system-level analysis, 2) automation-level analysis, 3) hardware-level analysis, and 4) software-level analysis.



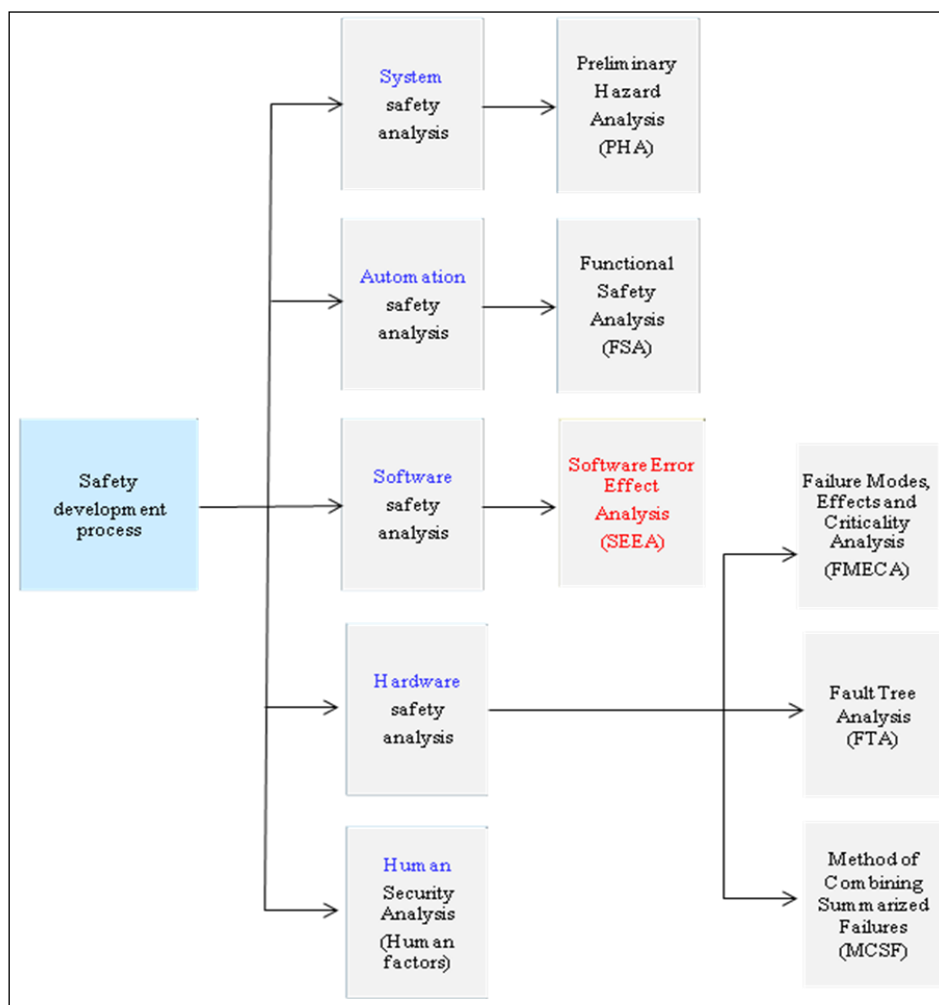
**Figure 1.** Activities for achieving safety.

Each level of analysis has one or more safety methods (Figure 2):

- 1) At the system level, the main method is the “Preliminary hazard analysis” (PHA) method. 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 results of this analysis are used to define the requirements and criteria for (high level) safety of the system, to establish the framework for the demonstration of safety as well as the outlines of downstream safety analyzes (Functional safety analysis, Hardware Safety Analysis and Software Safety Analysis).
- 2) At the level of automatism, a method known as “Functional safety analysis” (FSA). 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 (low level) safety criteria for the

design of the system and the realization of hardware and software safety equipment. They also impose safety criteria related to the sizing, operation and maintenance of the system. FSAs can highlight unsafe scenarios that require specification recovery and system design.

- 3) At a software level, it is a question of carrying out several methods related to Software Safety Analysis (SSA). The SSA is generally based on the SEEA method as well as on critical code reads.
- 4) At the hardware level, several safety methods relating to Hardware Safety Analysis (HSA) need to be established. The HSA focuses on electronic boards and interfaces defined of safety. This analysis implements two types of analysis:
  - An “inductive” analysis by analysis of failure modes, their effects and their criticality (AFMEC) generally completed by a method of summary failure combinations;
  - 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 functional safety analysis (FSA). This deductive analysis usually requires the use of the Cause Tree method.



**Figure 2.** Positioning of the SEEA method in the process of developing safety.

Indeed, all of these methods of safety analysis are based on two fundamental approaches, one of the “inductive” type and the other of “deductive” type. In the inductive approach, the reasoning goes

from the most particular to the most general; this leads to a detailed study of the effects of a failure on the system and its environment. In other words, inductive methods start from elementary events; either to look for consequences directly or to identify combinations of events that may have other than minor consequences PHA, AFMEC, SEEA are examples of inductive methods. In the deductive process, the reasoning goes from the most general to the most particular in such a way that, in the face of the failing system, the causes of the failure are deduced. The main deductive method is the Fault tree analysis (FTA). In practice, and in the face of a complex system, safety analysis requires experts in the field to implement an iterative safety development process involving both inductive methods and deductive methods. In the design, development and operation of a rail transport system, all the actors involved (infrastructure manager, railway undertakings, manufacturer, certification body, national safety authority, investigative body technical, etc.) use one or more safety methods to identify hazardous elements and equipment, hazardous situations, the causes of hazards, potential accidents, and the severity of consequences that would result.

After briefly presenting the main methods of safety analysis involved in the design and development of a rail transport system, the following section will be devoted to SEEA method, which is the subject of the study presented in this paper.

### 2.3. *Software errors and effects analysis (SEEA)*

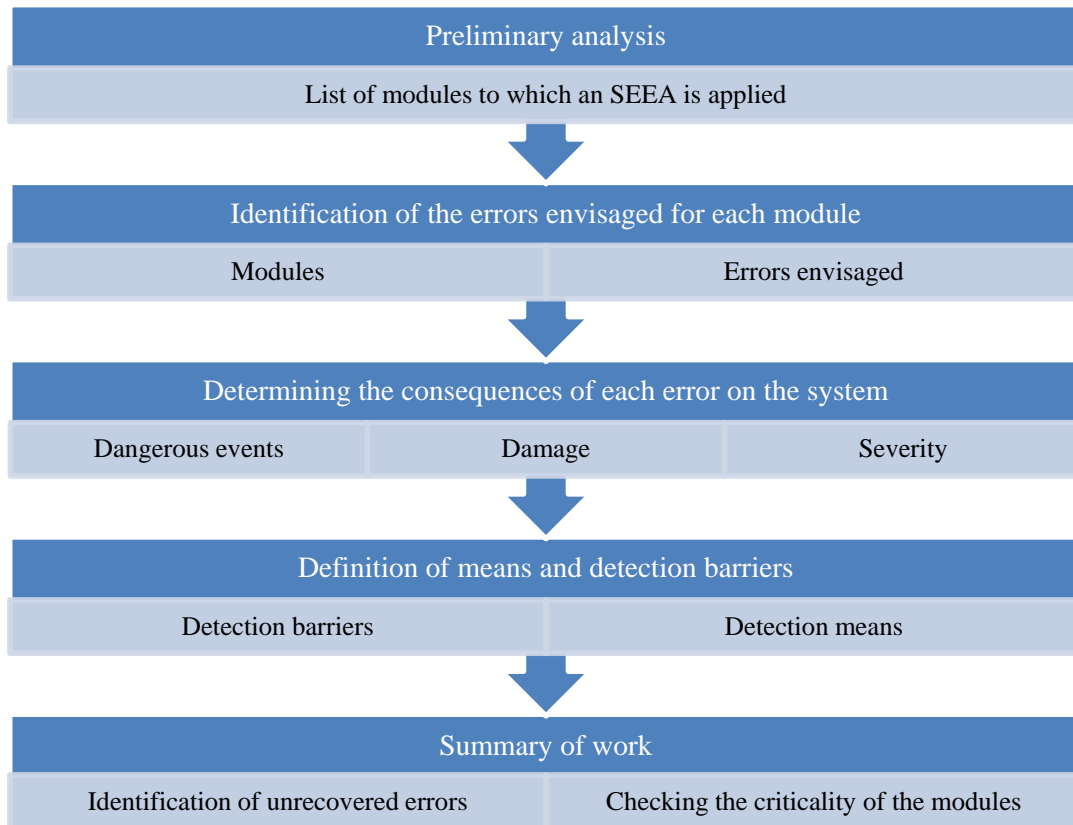
#### 2.3.1. Purpose of the SEEA method

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. According to the French standard NF F 71 013, a SEEA is carried out according to the following three stages (Figure 3) [2,3]:

- 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.

Depending on the systems studied, a SEEA form may be more or less complete, more or less close to standard NF F 71 013.

- 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.



**Figure 3.** Procedure for elaboration of an SEEA.

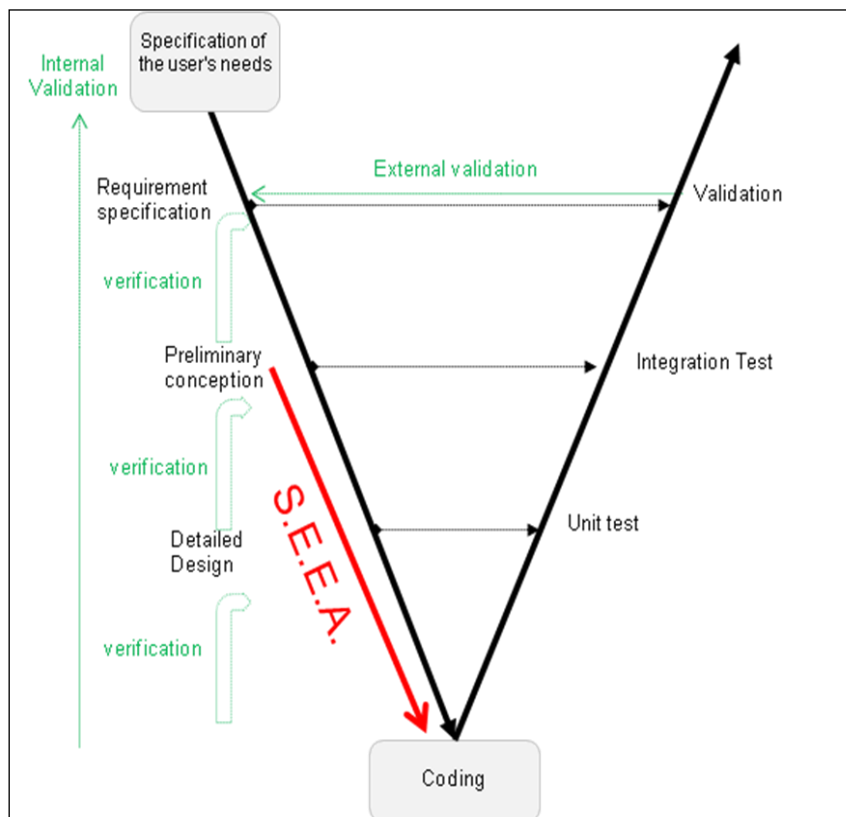
### 2.3.2. Place of the SEEA method in the development cycle of a project

Generally, in the development cycle of a project, there are three main levels of development: System level, hardware level, and software level. Figure 4 illustrates only activities related to the third level of the software, which assumes the following, seven chronological phases: requirements specification, preliminary design, detailed design, coding, unit test, modules integration test and finally the validation phase of the software functions. Thus, the sequence of a set of activities implemented in a specific order to achieve software is called a development cycle or life cycle of software. The main features of this approach are:

- The life cycle of the software is usually synthesized by a “V” diagram comprising a descendant branch for the specification and design of the software and an ascending branch for the integration and validation of the software;
- The chronology of the works is sequential, since each phase is conditioned by the completion of the previous phase;

- The transition from one phase to the next is conditioned by a verification step that we call “internal validation”;
- During these phases, the nature of the work to be done and the actors involved are clearly identified;
- At these different phases correspond to increasing levels of definition. It is an approach whose phases represent different levels of abstraction: from the general to the particular;
- During the downstream phases (specification and design), the plans and test scenarios necessary for the upward phases (integration and validation) are also prepared;
- In the ascending branch, during the last validation phase, it must be ensured that software project meets the requirements and criteria imposed during the specification phase. These verification steps are generally referred to as “external validation” overall.

All safety studies and methods are carried out in parallel with the various activities of developing a rail transport system project. Each of these activities corresponds, in fact, to one or more best suited safety analysis methods. As far as the SEEA method is concerned, it is usually developed during the descending branch of the software development cycle (Figure 4). The French standard AFNOR NF F 71-012 recommends starting them in the preliminary design phase when the safety software elements are identified, and taking them into account during the detailed design and coding phases. SEEA are often used by European railway companies. Indeed, the SEEA are recommended by the SNCF, highly recommended by SNCB (Belgian National Railway Company), the FS (Ferrovie dello Stato), the Italian company Ansaldo, and the LUL (London Underground Limited) and required by British Railways and Alstom.



**Figure 4.** The place of the SEEA method in the software development cycle.



### 2.3.3. Contributions and limits of the SEEA method

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.

## 3. Objective of the study and envisaged approach

### 3.1. Objective of the study

The completeness and consistency of the safety analysis of a rail transport system is essentially based on the know-how, intelligence and intuition of the human expert. However, a careful study of the reasoning mechanisms of the expert, his problem-solving strategies and heuristics, shows that he mainly uses symbolic, evolutionary and qualitative data and that he simultaneously uses inferences and modes reasoning of the inductive, deductive and reasoning based on the analogy and similarity between previous situations and the new risk situation. This has led us to use Artificial Intelligence techniques to systematize the SEEA approach and consequently to reinforce conventional methods of safety analysis. The purpose of this research is to develop methods and tools to lighten the task of the safety review specialist and in particular the certification experts. The approach followed is centered on the use of knowledge acquisition techniques and automatic symbolic learning and in particular CBR. The development of a SEEA support tool was motivated by various findings revealed by the problem identification and specification phase. The main ones are [4]:

- The need to rationalize and automate the classical SEEA approach;
- The need to improve the quality of accident risk analysis by archiving, formalizing and disseminating the know-how of the builder and safety experts;
- The difficulty of synthesizing and exploiting the considerable amount of historical knowledge involved in the SEEA's carried out for guided rail transport systems already certified and put into service in France;
- The desire to judge the comprehensiveness of the SEEA's proposed by the manufacturer as part of the review of the safety studies of a new rail transport system.

These reasons have guided us towards the development of a CBR-based tool for assisting in the analysis and review of the completeness and consistency of a new SEEA. More precisely, this tool must allow:

- To suggest risks and/or solutions not taken into account during the initial analysis;
- To help find the most appropriate solutions or preventive measures to guard against a particular risk;

- To propose a common risk analysis database of the SEEA for the various actors involved in the development of a rail transport system.

This base is indispensable especially when the studied system bears a resemblance to existing systems and the experience is lacking. It can be further enriched and updated by the various stakeholders. The main objective of this database is to sustain the experience and know-how in the analysis of risks and errors related to SEEA. The solutions chosen to design and implement this tool involve the following four main steps [4]:

- Formalization and structuring of knowledge with a view to identifying a representation and acquisition model for SEEA based on manufacturers' safety files;
- Collection of knowledge and more precisely risks and solutions adopted. In this step, we exploit the formalism elaborated in the previous step to archive typical cases;
- Creation of a knowledge base covering all SEEA for already certified transport systems;
- Exploitation of the database to help judge the completeness of a SEEA of a new system and in particular the safety of critical software.

The following paragraph presents the approach adopted for the acquisition of railway safety knowledge.

### *3.2. Approach retained for acquiring safety knowledge*

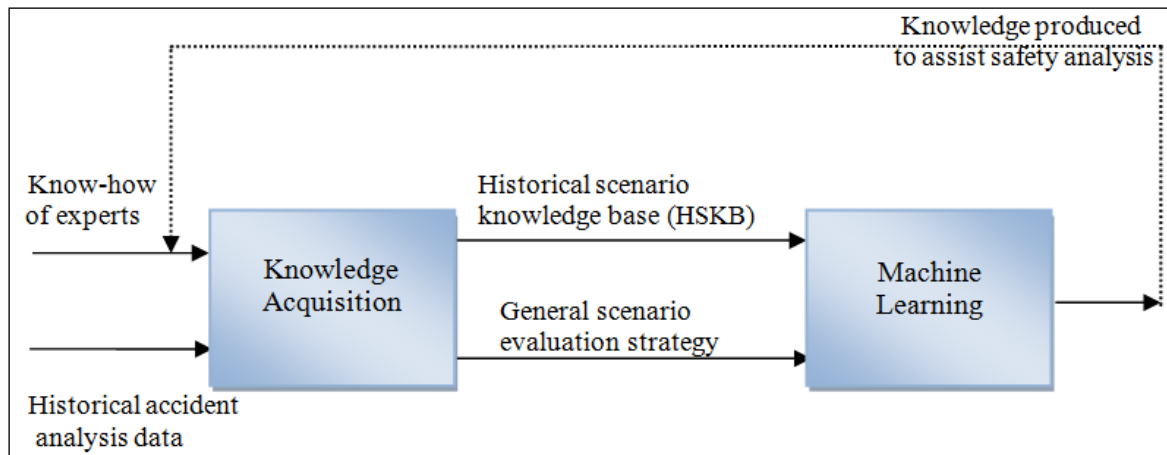
Knowledge acquisition was recognized as a bottle neck from the first appearance of expert systems, or more generally KBS. It is still considered to be a crucial task in their creation. Two main participants are involved in KA: The expert, who possesses know-how of a type which is difficult to express, and the "knowledge engineer" who has to extract and formalize the knowledge which is related to this know-how, which as far as the expert is concerned is usually implicit rather than explicit. This time-consuming and difficult process is nevertheless fundamental to the creation of an effective knowledge base. Some work suggests viewing the design of a KBS as a process of constructing a conceptual model, on the basis of all the available sources of knowledge (human or documentary) which relate to solving the problem. In this context KA is perceived as a modeling activity. Other research stresses the benefits of methods which guide the knowledge engineer in the transfer/modeling process. Tools and techniques are used to provide assistance with verbalization, interviews with experts and document analysis. Currently available KA techniques mainly originate in cognitive psychology (human reasoning models, knowledge collection techniques), ergonomics (analysis of the activities of experts and the future user), linguistics (to exploit documents more effectively or to guide the interpretation of verbal data) and software engineering (description of the life cycle of a KBS). A survey of state of the art research in the domain of knowledge acquisition made it possible to select a method for developing a KBS for aid in the analysis of safety for guided rail transport system. This method showed itself to be useful for extracting and formalizing historical safety analysis knowledge (essentially accident scenarios) and revealed its limits in the context of the expert safety analysis, which is particularly based on intuition and imagination. In general, current knowledge acquisition techniques have been designed for clearly structured problems. They do not tackle the specific problems associated with multiple areas of expertise and the coexistence of several types of knowledge and it is not possible to introduce the subjective and intuitive knowledge which is related to a rapidly evolving and unbounded field such as safety. Although cognitive psychology and software engineering have

produced knowledge acquisition methods and tools, their utilization is still very restricted in a complex industrial context.

One possible way of reducing these constraints is combined utilization of knowledge acquisition and machine learning techniques. Experts generally consider that it is simpler to describe examples or experimental situations than it is to explain decision making processes. Introducing machine learning systems which operate on the basis of examples can generate new knowledge which can assist experts in solving a specific problem. The know-how of experts depends on subjective, empirical, and occasionally implicit knowledge which may give rise to several interpretations. There is generally speaking no scientific explanation which justifies this compiled expertise. This difficulty emanates from the complexity of expertise which naturally encourages experts to give an account of their know-how which involves significant examples or scenarios which they have experienced on automated transport systems which have already been certified or approved. Consequently, expertise should be updated by means of examples. Machine learning can facilitate the transfer of knowledge, particularly when its basis consists of experimental examples. It contributes to the development of the knowledge bases while at the same time reducing the involvement of cognitive scientists. Expertise in a domain is not only possessed by experts but is also implicitly contained in a mass of historical data which it is very difficult for the human mind to summarize. One of the objectives of machine learning is to extract relevant knowledge from this mass of information for explanatory or decision making purposes. However, learning from examples is insufficient as a means of acquiring the totality of expert knowledge and knowledge acquisition is necessary in order to identify the problem which is to be solved and to extract and formalize the knowledge which is accessible by customary means of acquisition. In this way each of the two approaches is able to make up for the shortcomings of the other. In order to improve the process of expertise transfer, it is therefore beneficial to combine both approaches in an iterative knowledge acquisition process. Our approach has been to exploit the historical scenario knowledge base (historical case base of SEEA) by means of learning with a view to producing knowledge which could provide assistance to experts in their task of evaluating the level of safety of a new system of transport. The approach which was adopted involved the following two main activities (Figure 5) [5–7]:

- Extracting, formalizing and storing hazardous situations to produce a library of standard cases which covers the entire problem. This is called a historical scenario knowledge base. This process entailed the use of knowledge acquisition techniques;
- Exploiting the stored historical knowledge in order to develop safety analysis know-how which can assist experts to judge the thoroughness of the manufacturer's suggested safety analysis. This second activity involves the use of machine learning techniques.

After having presented the approach adopted for the development of a tool to assist in the analysis and evaluation of the security of critical software used in the field of rail transport safety and in particular the improvement of the method SEEA, it is now necessary to present a bibliographic study on artificial intelligence techniques in order to position our work in relation to existing works.



**Figure 5.** Processes of safety knowledge acquisition [5, 6 and 7].

#### 4. Artificial intelligence: Literature review

The intellectual process by which a human operator evaluates a situation, predicts an event or makes a decision is often difficult to model in the form of reliable and definitive algorithms. This difficulty can be partially overcome by using Artificial Intelligence (AI) techniques. In recent years the considerable development of AI techniques has made it possible to overcome the inadequacy of traditional computing. AI aims to study and simulate human intellectual activities and strives to create machines capable of “intelligent” behavior. Artificial Intelligence aims ambitiously to equip the computer with some of the faculties of the human mind: To learn, to recognize, to reason, etc. The ability to understand natural language and the ability to reason is the keystone of AI. Nowadays, almost everyone is excited about AI and Machine Learning and a hundred new conferences appear all over the world for the year 2019: In San Francisco: AI and Future of Work, in China: Global Symposium on AI, London: Summit on In-Depth Financial Learning, London again: AI and Big Data, New York: AI Conference, California: International Conference on Machine Learning, Hong Kong: Summit on AI, Montreal-Canada: Summit on Deep Learning, etc.

##### 4.1. Knowledge-based systems (KBS)

The first results of the research carried out in this way concern expert systems or KBS. They have emerged as decision support tools capable of replicating some of the intellectual tasks usually performed by human experts. The capacity to exploit and especially to capitalize and sustain the experience gives the KBS a power of information and decision in order to guide the non-specialist users. Since the 1970s, the KBS have made a remarkable entry into the industry; they are no longer considered rare objects from research laboratories. However, they very rarely achieve the performance of the human expert and are often poorly adapted to the real needs of end users. This is due to the difficulty of extracting the necessary expertise from one or more experts in the field and to represent this knowledge without distortion to arrive at a cognitive model of the expert. In addition, this is a crucial phase of manual filling of KBS knowledge base (KB). Capturing the knowledge to keep it in the KB of an expert system is a complex and time consuming task and often in material and human resources. Several research works have evoked the problem of collecting and formalizing

the knowledge manipulated by the problem solving expert. The expert may have great difficulty in explicitly describing the steps in the reasoning he uses to make decisions. This requires a long process of reflection that will allow him to explain the unconscious part of his approach. This difficult and sometimes painful task depends on the success of a KBS project. Like many authors, we consider this task as the bottleneck encountered in the development of a KBS. Indeed, given the complexity of the knowledge of the expert and the difficulty of the latter to explain its mental processes, the knowledge extracted may often be inaccurate, incomplete or even incoherent. Various researches in AI are thus carried out to apprehend this problem of transfer of expertise. There are two major independent research activities today: Knowledge Acquisition (KA) [8–10] and Machine Learning (ML) [11–15].

#### *4.2. Knowledge acquisition (KA)*

The design of a knowledge base requires the extraction, analysis, structuring and formalization of the know-how of a domain that is accessed through one or more individuals, qualified experts. Therefore, the transfer of this expertise raises the following delicate questions: Who really holds the expertise, how can we access it, how to extract it, how to formalize it without distorting it, which representation to choose? How to validate and maintain the knowledge collected? Various researches are being conducted to better understand these problems inherent to the acquisition of knowledge and the design of a KBS. Methods, techniques and tools for the acquisition of knowledge are now accessible to the cognitive engineer (Knowledge Engineer) and to the expert and offer a methodological framework for the development of a KBS. These techniques, often inspired by work in cognitive psychology, make it possible to access and make as explicit as possible the knowledge manipulated by the domain expert. Several terms are proposed in the literature to translate the word “collection” of knowledge. We speak of techniques of elicitation, enunciation, extraction, externalization, capture or solicitation of knowledge. These techniques are generally referred to as “cognitive” methods, “manual” methods or “empirical” methods. The possible techniques for extracting knowledge have been studied and presented by many authors [8–10].

#### *4.3. Machine learning (ML)*

##### *4.3.1. Objectives of machine learning*

The notion of learning is a very general term that describes the process by which the human being or the machine can increase his knowledge. To learn is therefore to reason, to discover analogies and similarities, to generalize or particularize an experience, to take advantage of past failures and errors for later reasoning. The new knowledge is used to solve new problems, to carry out a new task or improve performance of an existing task, to explain a situation or predict behavior [13]. The areas of human activity are increasingly complex and involve amounts of information that the human mind synthesizes with difficulty. Extracting from this mass of data relevant and useful knowledge for explanatory or decisional purposes is the main objective of machine learning (ML). ML is an important branch of research in the field of AI. The birth of this discipline dates back to the 1960s and the most spectacular result was obtained at that time by the American A. SAMUEL. The latter has designed a program for the named checkers game

“CHECKERS” who, memorizing a lot of shots, constantly improved his strategy and eventually reached the level of a champion in this discipline. The principle of learning was born: To learn is to perfect one’s knowledge and improve one’s performance by taking advantage of past failures. In the 1970s, a new approach to learning emerged: The AI approach, which aims for explicability in the knowledge base formed. Thus, apprenticeship, which at first was only an interesting idea, has now become an indispensable discipline for the progress of several industrial systems.

This discipline is regarded as being a promising solution for knowledge acquisition aid and attempts to answer certain questions [11]: How can a mass of knowledge be expressed clearly, managed, added to and modified? Machine learning is defined by a dual objective: A scientific objective (understanding and mechanically producing phenomena of temporal change and the adaptation of reasoning) and a practical objective (the automatic acquisition of knowledge bases from examples). Learning may be defined as the improvement of performance through experience. Learning is intimately connected to generalization [13]: Learning consists of making the transition from a succession of experienced situations to knowledge which can be re-utilized in similar situations. Three types of problems are raised for each of the main learning techniques [11]. The first of these is grouping (which is termed classification in data analysis): Given a certain mass of knowledge, how is it possible to discover links between the different items in order to group them into meaningful and simpler sub-groups? The second problem (discrimination) is that of learning classification procedures: With a given set of examples of concepts, how is it possible to find a method which provides effective recognition of each concept? The third problem is that of generalization: How is it possible, on the basis of concrete examples of a situation, to find a formula which is sufficiently general to describe the situation in question and how is it possible to explain the descriptive ability of this formula?

ML has attracted increasing interest in recent years, as evidenced by the impressive number of publications and conferences it is the subject of. ML’s efforts to address clustering, discrimination and generalization of objects have resulted in a wide variety of methods, techniques, algorithms and systems. Nevertheless, this abundant literature makes it difficult to perceive the field, given the ambiguity of its vocabulary and the absence of rigorous reference definitions.

#### 4.3.2. Principle of operation of an automatic learning system

Very schematically, for a learning system to produce results, it must have essentially 1) a theory of the domain, 2) a set of learning examples, 3) possibly classes of objects, 4) one or more reasoning mechanisms (induction, deduction, abduction, analogy), 5) knowledge and rules necessary for the evaluation of the results produced or learned by the system (in the context of supervised learning) and finally, 6) knowledge for processing incomplete (noisy) data.

Domain theory refers to knowledge of the field of application also known as “prior” knowledge or “background” knowledge. It expresses any explicit knowledge of the application domain such as the type and scope of the descriptors and their relationships. The domain theory generally includes the description language of the learning examples and the learning parameters. The existence of a learning set presupposes the definition of a “source” language to represent the examples and a “target” language to formalize the knowledge learned. The choice of these languages results from a prior analysis of the nature of the available data. Learning examples are the experimental set on which the learning process will operate to generate new knowledge, concepts, characterizations or

rules. To arrive at relevant knowledge, the set of learning examples must be non-noisy and sufficiently representative of the field of application. An example is called positive if it belongs to a class of objects or negative if it does not belong to this class and is actually a counterexample. If a learning algorithm makes it possible to generate rules or concepts from experimental examples, the fact remains that the quality of the knowledge learned depends to a large extent on the quality of the database of examples (correct information complete, consistent, rich, sufficient number of examples and descriptors). Machine learning is particularly sensitive to the relevance of the available data. The control of this quality relies in particular on the acquisition and the use of complementary knowledge to reduce the diffuse noise in the examples.

In the end, based essentially on a theory of the domain, an experimental set of examples, the learning process based on reasoning mechanisms (induction, deduction, abduction, analogy) produces strategies for solving problems such as explanation, prediction, classification or recognition. This knowledge is usually expressed as decision trees, production rules, or concept hierarchies. When the learning examples covering the field of application are considerable in number and cannot be exhaustively identified with the experts, it is highly desirable to use incremental learning. A learning system is said to be incremental if it has the capacity to evolve knowledge learned in a previous cycle, without having to reprocess all the examples collected.

#### 4.3.3. Different modes of reasoning in machine learning

The machine learning mechanism is based on four modes of reasoning or inference: Induction, deduction, abduction and analogy Grundstein defines these terms through an example:

- ◆ Deduction proceeds from a rule and a fact to obtain a result:
  - Rule: All beans in this bag are white;
  - Fact: These beans are pulled from this bag;
  - Result: These beans are white.
- ◆ Induction leads to the rule from a fact and a result:
  - Fact: These beans are from this bag;
  - Result: These beans are white;
  - Rules: All beans in this bag are white.
- ◆ Abduction leads to the fact starting from the rule and from a result:
  - Rule: All beans in this bag are white;
  - Result: These beans are white;
  - Fact: These beans are pulled from this bag.

Example 1:

- Deduction: “Socrates is a man, men are mortal  $\rightarrow$  Socrates is mortal”.
- Abduction: “Socrates is mortal, cats are mortal  $\rightarrow$  Socrates is a cat”.
- Induction: “I see a French woman, she is red  $\rightarrow$  all French are red”.

In the context of “abduction”, we reason from the effects to find the probable causes, while in the context of the “deduction” we reason from the causes to deduce the possible effects.

Example 2:

Knowing that  $\forall x [\text{Cloud}(x) \rightarrow \text{Rain}(x)]$

- It’s an abduction to predict clouds when you see the presence of rain;

- It is a deduction to predict that there will be rain when we see the presence of clouds.

If we consider the famous Modus Ponens:

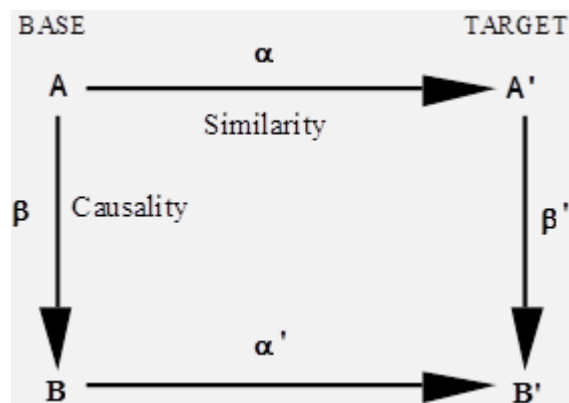
- Deduction is a safe mode of inference: From A and  $A \rightarrow B$ , deduce (or infer) B;
- Abduction is characterized by an inversion of the Modus Ponens;
- Induction seeks to determine general cause and effect relationships from specific facts.

These three mechanisms of reasoning can be more formally represented:

- Deduction: From A and  $A \rightarrow B$ , we deduce B;
- Abduction: From B and  $A \rightarrow B$ , we “abduct” A;
- Induction: From  $A(z) \rightarrow B$  and from  $A(w) \rightarrow B$ , we “induce”  $A(x) \rightarrow B$ .

- ◆ Analogy, for its part, derives from fact: It consists in inferring that a true property of an object can also be so for another having similarities with the first. The plausibility of the conclusion depends on the similarity between these two objects. Induction, abduction, and analogy are described as non-rigorous reasoning. Indeed, these mechanisms apply to certain knowledge but lead to conclusions that are only plausible. Analogy is used in practice to understand or interpret new situations from previously memorized situations. The analogy combines both the notion of similarity (or resemblance) and the notion of causality. More formally, an analogy includes a source situation of the form (A, B) and a target situation of the form (A', B'). There are similarity (and dissimilarity) relations between A and A', respectively B and B', as well as dependence relations, generally of a causal nature, between A and B, respectively A' and B'. To make an analogy is to start from an incomplete schema similar to this one of Figure 6 and to complete it by combining similarity and causality. In summary, learning by analogy first involves recognizing similarities between the target concept to be learned and a known source concept and then determining that relevant characteristics can be transferred from the “source” to the “target”.

The CBR research only looks for similarities or proximity relations between past situations and the current situation.



**Figure 6.** Principle of reasoning by analogy.

#### 4.4. Case-based reasoning (CBR)

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



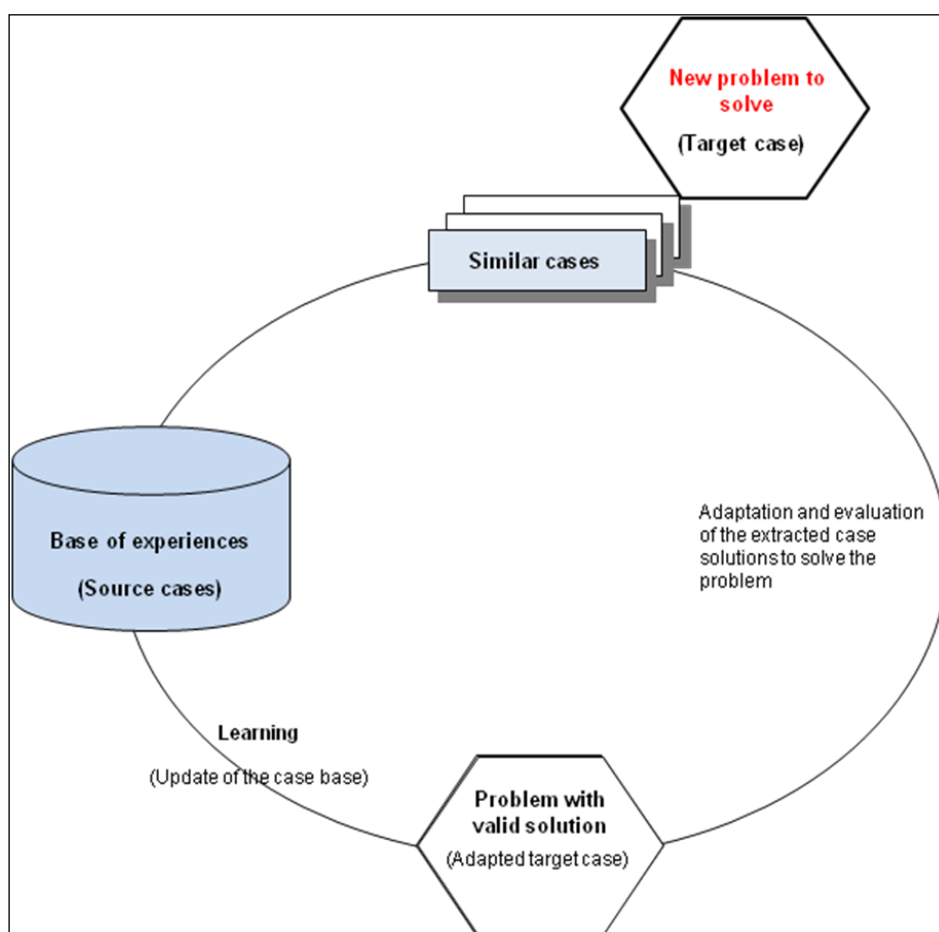
solving everyday human problems. Indeed, all human reasoning is generally based on past cases lived personally. The CBR considers reasoning as a process of remembering a small set of practical situations: The cases, it bases its decisions on the comparison of the new situation (target cases) with the old (reference cases). The general principle of CBR is to treat a new problem (target case) by remembering similar past experiences (source cases). This type of reasoning rests on the assumption that if a past experience and new circumstances are sufficiently similar, then everything can be explained or applied to past experience (source cases) and remains valid when applied to the new situation which represents the new problem to solve. For example, in the field of technical or medical diagnostics, the expert in the field, faced with the symptoms observed, he often proceeds by analogical reasoning by referring to past historical cases to quickly explore and search for the causes of a risk of accident or illness (for the doctor) in order to propose a remedy for this new undesirable situation. CBR is an approach to problem solving that emphasizes the role of prior experience during future problem solving (i.e., new problems are solved by reusing and if necessary adapting the solutions to similar problems that were solved in the past). 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 (Figure 7):

- The establishment of indexes (or indexing): We assign certain indexes to the new problem (target case) to characterize it. These indexes are formed from the information extracted from the new problem and will be used to search for similar cases. 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).
- 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 generally 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.
- Reusing cases: Two possibilities arise. 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; either

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.

- Revision: After having associated the previous solution with the new situation, it is necessary to test the new solution in the real world by having recourse to a simulation or to the expert of the field and, if necessary, to revise the adopted solution.
- Learning: Once the solution is adapted to the target problem, it is advisable to archive and store this new experience as a new case in the initial database (source cases). This is learning and updating step to enrich the experience in a given field. Thus, by analogy to inductive automatic learning, the CBR starts the learning cycle with a set of cases or learning examples; it then proceeds by an approach of generalization of these examples (inductive learning) by identifying the common points between a recovered case of the base of examples and the target problem.

The work of Aamodt [16], Harmon [17], Kolodner [18,19], Leake [20], Mott [21], Pinson [22] and Slade [23] provide a fairly complete retrospective of the evolution of case-based reasoning research (CBR).



**Figure 7.** Cycle case based reasoning [4].

#### 4.5. The applications of artificial intelligence

In addition to KBS, Knowledge Acquisition (KA) and Machine Learning (ML) previously mentioned, artificial intelligence implements several other methods and techniques such as neural networks (NN) also called deep learning, genetic algorithms (GA), pattern recognition, which is often associated with image processing, fuzzy systems based on the fuzzy set theory proposed by Lotfi Zadeh in 1965, Big Data Analytics (BDA), reasoning based on case (CBR). For those who are involved or interested in the latest AI technologies, the annual conference on Applied Artificial Intelligence will take place on April 18, 2019 in San Francisco. It will focus on the latest trends and the future impact of CEW applications and commercialization in many sectors, including transportation, logistics, health, energy, financial technologies, the future of work (FoW), the Internet of Things (IoT), and cyber security.

##### 4.5.1. Examples of AI applications in the field of rail transport

In recent years and in the field of land and air transport, researchers and experts in the field have become increasingly interested in the application of artificial intelligence techniques to solve certain problems of aid. the decision, such as the diagnosis of transport equipment, the management of maintenance operations, the analysis of driver behavior, the prediction of the deterioration of transport infrastructure, the planning and forecasting of traffic demand, control of traffic signals, control of air traffic, etc. For example, machine learning has been used for rail maintenance forecasting [24], Expert Systems (fuzzy knowledge based) for rail traffic control [25], deep learning for the detection of lateral defects of the railroad [26] and neural networks for the detection of defects on the surface of rails [27]. As for Big Data Analytics (BDA), it has been used in particular to identify trends, discover relationships, implement predictive analysis and to give meaning to images, data flows and various information. In railway transport applications, the BDA can be of a beneficial contribution in view of the large amounts of data generated by the transport system from sensors installed on the tracks, on the wagons or from the signaling equipment, monitoring and inspection equipment, communication systems, train monitoring systems, etc. A BDA can examine the collected data set in order to obtain useful information to explain for example the potential causes of degradation of the operation, the failure of certain track components and possibly safety equipment. The BDA therefore presents the main characteristics required by rail transport experts to monitor the overall condition of the infrastructure, optimize and plan maintenance operations, manage the risks of accidents and potential incidents and consequently improve safety of the transport system. As an example, we can mention the work on exploiting data relating to operation, maintenance and railway safety [28], decision-making on rail maintenance [29], engineering and the management of railway applications [30], the improvement of call reporting systems [31], the implementation of a predictive approach to the safety and maintenance of personnel [32], and Siemens on the use of Big Data to build the Internet of trains [33].

The main objective of our study is to develop a tool for analyzing and evaluating SEEA based on CBR. The following section presents a literature review on CBR's contribution to rail transport safety.

#### 4.5.2. Examples of CBR applications in the transport sector

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. CBR is attracting more and more attention from researchers and experts in the rail transport sector. This therefore argues for the need to review recent research in this area with a view to providing a comprehensive review of the major recent applications in the context of rail transport. CBR is a well-established field of research based on artificial intelligence techniques and in particular machine learning, as evidenced by the 27th International Conference on Case-Based Reasoning (ICCBR) held in Stockholm, Sweden from July 10 to 12, 2018. This mode of reasoning, which is based on the notion of similarity, focuses primarily on problem solving based on experience. It is a cognitive process of human reasoning that relies heavily on how people acquire a new skill based on their past habits and experiences. CBR means using and exploiting old experiences to understand, explain, interpret or solve new situations similar to similar past situations. CBRs are increasingly used in industrial applications such as technical diagnostics, medical diagnostics, image processing, law, design, planning, and so on. In the field of transportation, 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 [34]. In the road transport sector, the application of CBR is numerous: Transport planning [35], management of traffic flows [36,37], control of urban intersections to avoid road congestion [38], the analysis of road collisions [39], the improvement of traffic in urban intersections by developing new signaling plans [40], the control of traffic flow at intersections (traffic control systems (TCS)) [41], the diagnosis of the driver's stress level [42], or the modeling of the risk of driver fatigue [43]. Finally, in the rail transport sector, studies include the diagnosis of locomotive failures [44], the recovery of incident reports [45], the prevention of rail operations incidents [46], the command of railway rescue (Emergency Relief Command) [47], analysis of safety risks related to the operation of the metro [48], automatic train conduction to reduce travel time and save fuel consumption [49] and finally the diagnosis of failures of the rail switching system [50].

### 5. Contributions with respect the state of the art

As mentioned in paragraph 3, safety experts and certification bodies face several obstacles to improving the safety level of rail transport systems, in particular the difficulty in synthesizing and exploiting considerable historical knowledge of SEEA (experience feedback) and the willingness to judge the completeness of the SEEA proposed by the manufacturer during the development of a new rail transport system. Thus, the need to rationalize the traditional approach of the SEEA, to improve the quality of accident risk analyzes and finally to assist the experts to judge the completeness of the SEEA and the adequacy of the protective measures considered, directed us towards the development of a tool based on the RBC allowing to suggest potential accidents and/or measures of protection or prevention most appropriate to protect oneself against a particular risk. In addition, we have also shown that improving the process of transferring rail safety expertise requires simultaneous use of knowledge acquisition and machine learning techniques.

However, the artificial intelligence approaches presented in Section 4.5. (The applications of artificial intelligence) cannot provide satisfactory answers to our research objectives. Indeed,

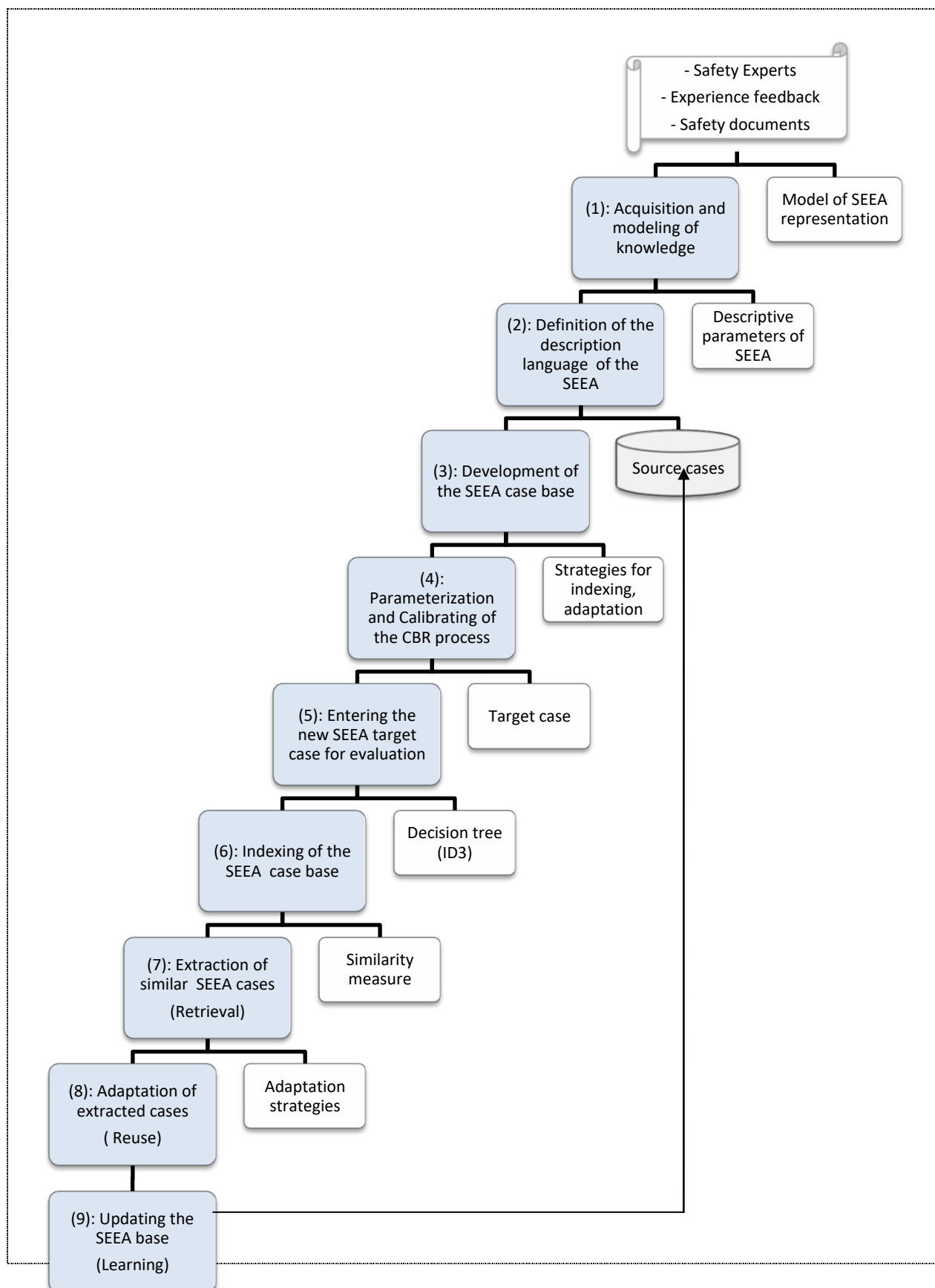
despite the undeniable interest of these approaches, to our knowledge, to date there are no applications of artificial intelligence to improve the security of critical software used in the rail transport sector and in particular tools to improve the SEEA method. Specifically, the bibliographic study (paragraph 4.5.2: Examples of CBR applications) 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.

## **6. Proposal for a method of assessment of critical software safety based on the CBR**

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 security experts in their SEEA document analysis and assessment tasks. The design and implementation of this tool required the following three major phases:

- Acquisition and modeling of knowledge related to SEEA. This analysis and abstraction stage resulted in the production of formalism for SEEA which takes account of the practices and our experience in the field of railway safety. This model is based on eight characteristic parameters: The investigated system, the investigated subsystem, the investigated module, the envisaged error (family, class, type), the safety criterion infringed by the error, the feared hazard, the type and severity of possible damage and finally the means of detecting the error and protecting against it.
- Using the above model we built up a library of 250 cases (examples). These historical examples of SEEA were drawn from two guided transport systems: MAGGALY and the TVM 430 for the Nord TGV.
- Development of the SAUTREL tool [51,52]. The mock-up has four main modules: A man/machine interface for inputting, updating and consulting knowledge relating to SEEA, a representation and acquisition module for SEEA sheets, a knowledge base containing 250 examples of SEEA (experience base), and a case-based reasoning process (implemented by the Recall software). The main components of this CBR process are a mechanism which indexes (or characterizes) target cases and a mechanism which finds similar cases (reference cases) and collects them together.

These three major phases of development of the tool to aid the assessment of the security level of critical software are detailed in Figure 8 in nine steps presented in the following paragraphs. As shown in Figure 8, 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 SEEA. Step 3 on the development of the database the SEEA made it possible to compile all the source cases.

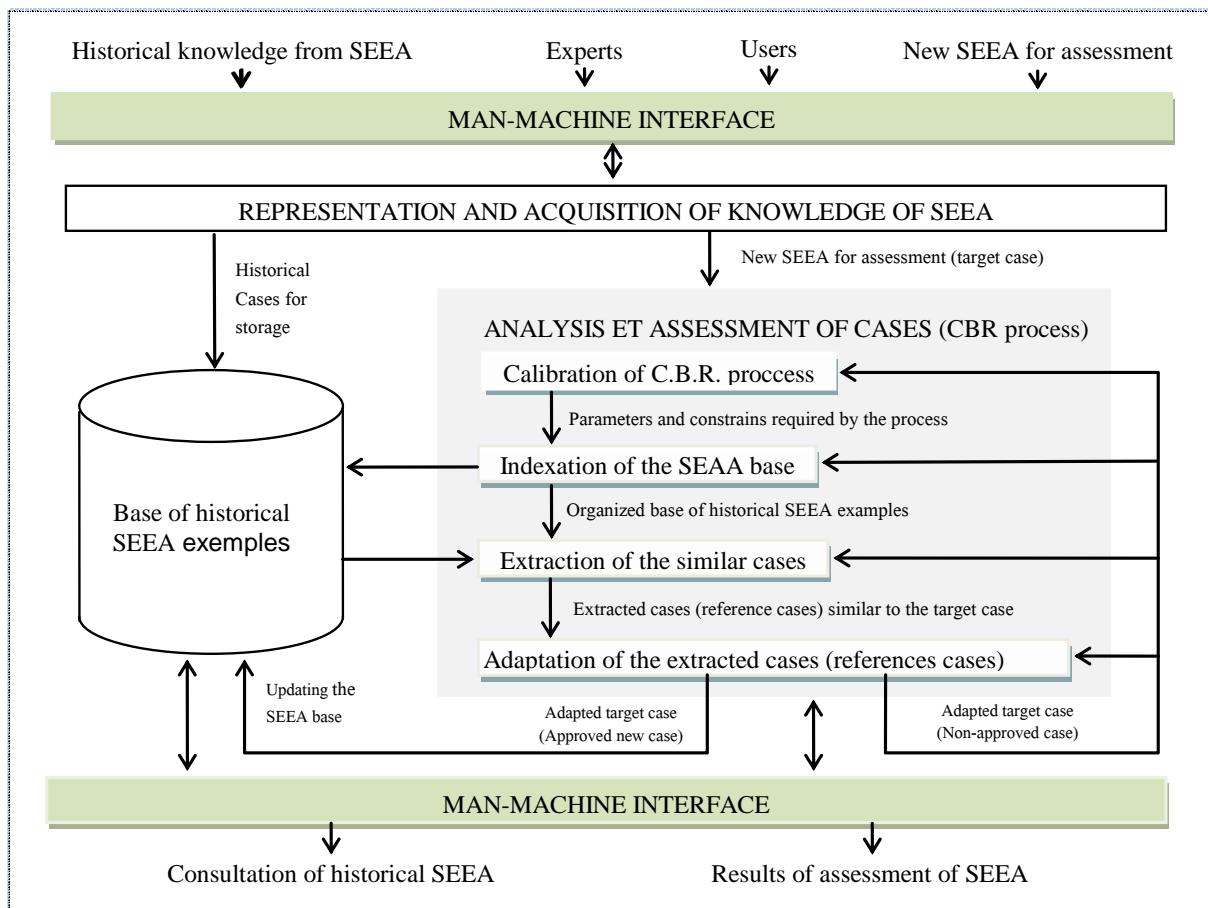


**Figure 8.** Approach for acquisition, modeling, capitalization and assessment of SEEA.

## 7. Example of the mock-up use

The mock-up (Figure 9) has been implemented using the ReCall software, marketed by ISoft firm, which generates CBR process. The following paragraphs show, through an example the use of this mock-up, which requires going through the nine following stages:

1. Acquisition and modeling of knowledge (Figure 10);
2. Definition of the description language of the SEEA examples (Figure 11);
3. Development the SEEA case base (Figure 12);
4. Parameterization and Calibrating of the CBR process (Figures 13, 14 and 15);
5. Entering the new SEEA target case for evaluation (Figure 16);
6. Indexing of the SEEA case base (Figure 17);
7. Extraction of similar SEEA cases (Figure 18);
8. Adaptation of extracted cases (source cases) (Figure 19);
9. Updating the SEEA base.

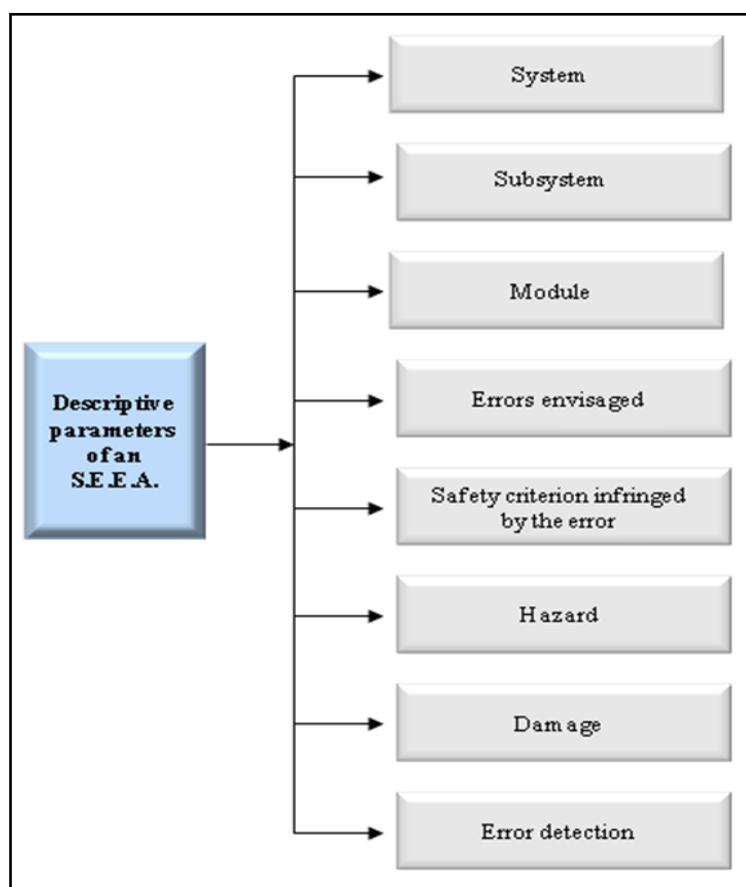


**Figure 9.** Functional architecture of the developed mock-up.

### 7.1. 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 documents SEEA. 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).



**Figure 10.** Descriptive parameters of an SEEA analysis.

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 (Figure 10).

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 Table 1. On the basis of this representation model of the SEEA forms, we have created a library of 224 typical cases. The development of the SEEA document representation model is organized around eight characteristic parameters (descriptors) (Table 1).

**Table 1.** Extract from the formalism elaborated for the representation of records SEEA.

SYSTEM			
MAGGALY-Line D	TVM 430- LGV Nord	Etc.	
SUBSYSTEM			
Train Edge	Train Floor		
MODULE			
Train location	Evacuation and door monitoring	Etc.	
ERRORS ENVISAGED			
<b>Error families</b>	<b>Classes of errors</b>	<b>Errors</b>	
Calculation error	Evaluation of an incorrect equation	Incorrect calculation	
Algorithm error	Instruction sequencing error	Forgetting a possible case during a test	
SAFETY CRITERION INFRINGED BY THE ERROR			
Criteria for anti collision	Activation of emergency braking (EB) in case of a route proposed by automatic piloting (AP) of unknown type or needle position mismatch	Etc.	
FEARED HAZARD			
Collision	Derailment	Maintaining High Voltage (HV)	Etc.
DAMAGE			
<b>Type</b>	<b>Severity</b>		
Individual	Collective	Level	Level 1    Level 2    Level 3
ERROR DETECTION			
<b>Detection barrier</b>	<b>Means of detection</b>		
No detectable	Beacon implementation specification		
Detectable by hardware barriers at the system level	Etc.		
Detectable by software barriers located in other modules			
Detectable by software barriers located in the module			
Etc.			

## 7.2. 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 (Table 1). 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. In Figure 11, the term “cardinal” of the attribute “criterion not

respected by the error” indicates that this attribute is multivalued. It can take between 1 and 4 values connected by an AND relation.

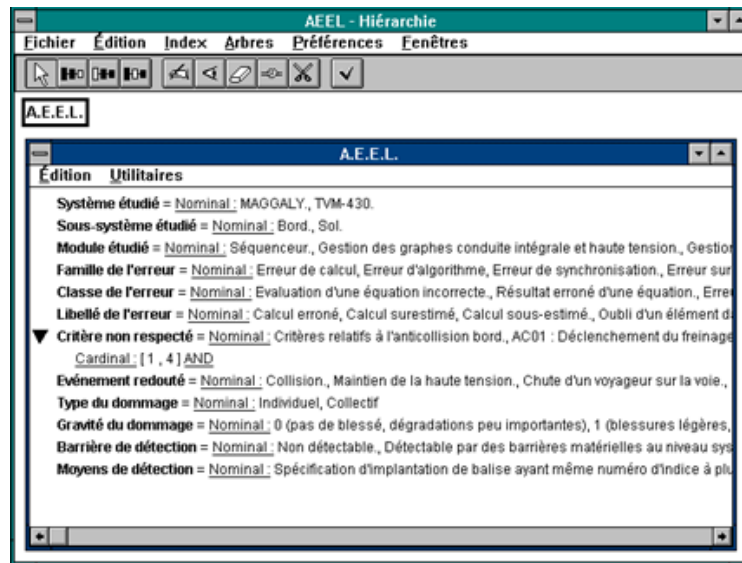


Figure 11. Definition of SEEA instances description language.

### 7.3. Developing the SEEA case base

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 (Figure 12). The symbol “?” means that an attribute is unknown and could not be populated by the domain expert. This is a missing value for the SEEA case. We present below how the ReCall software handles missing values.

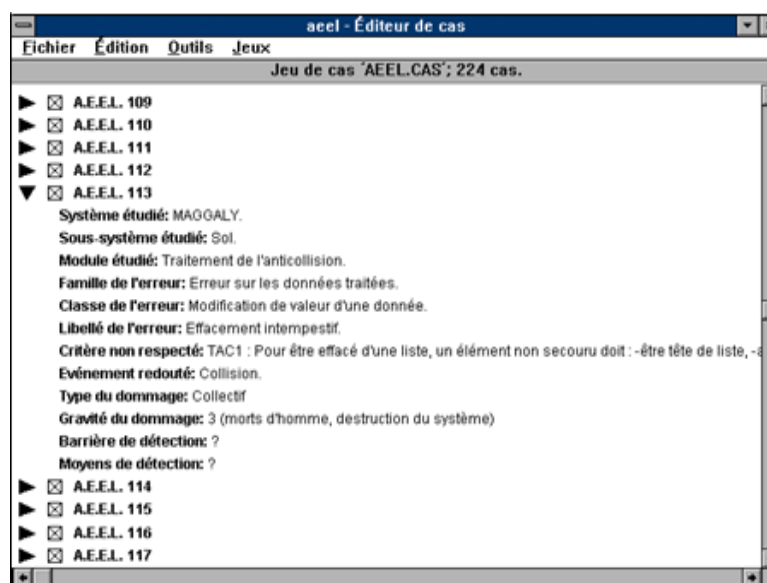
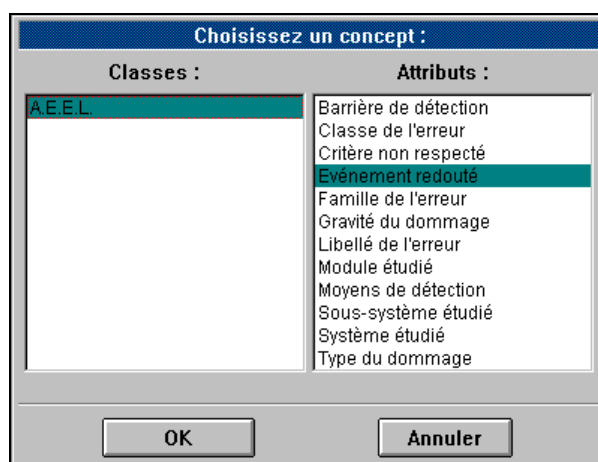


Figure 12. Construction of the SEEA base.

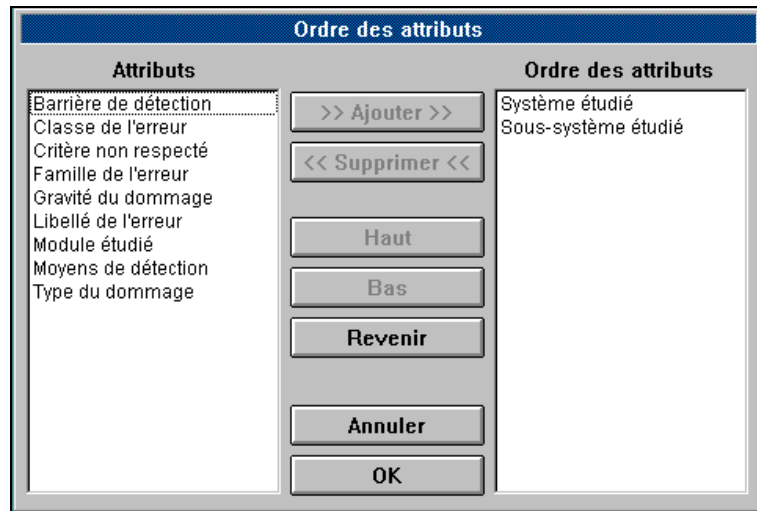
#### 7.4. Parameterization of the CBR process

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:

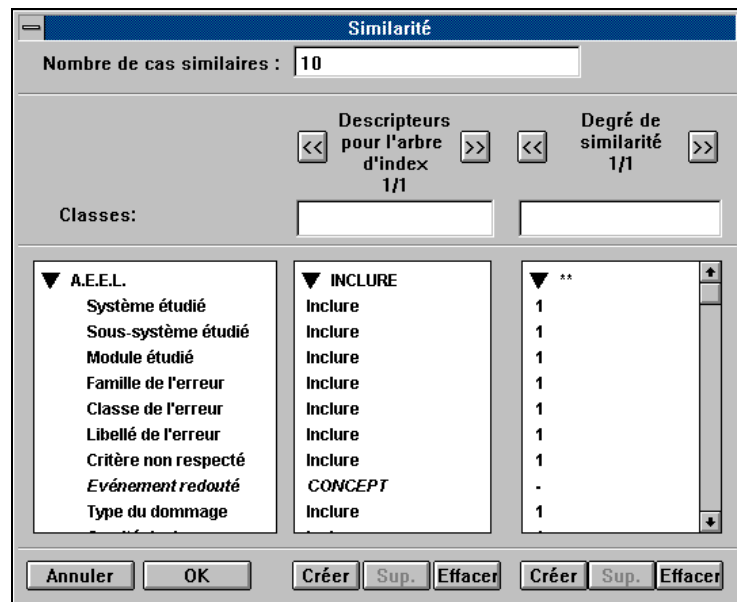
- The descriptor “concept” (Figure 13): The user must choose from all the descriptors the one that will represent the solution of the problem. This descriptor is called the “concept”. In our example, the descriptor “concept” is the descriptor “dreaded event”. The problem, meanwhile, will be characterized by all the other descriptors.
- Indexing strategies (Figure 14): 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 (Figure 15): 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: To date, the tool does not offer a real adaptation method, but allows the user to program his own methods by demons. 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:  $(100 \times \text{Descriptor weight 1} + 100 \times \text{Descriptor weight 2} + 0 \times \text{Descriptor weight 3})/3 = 66$ .



**Figure 13.** Choice of the attribute “dreaded event” as concept.



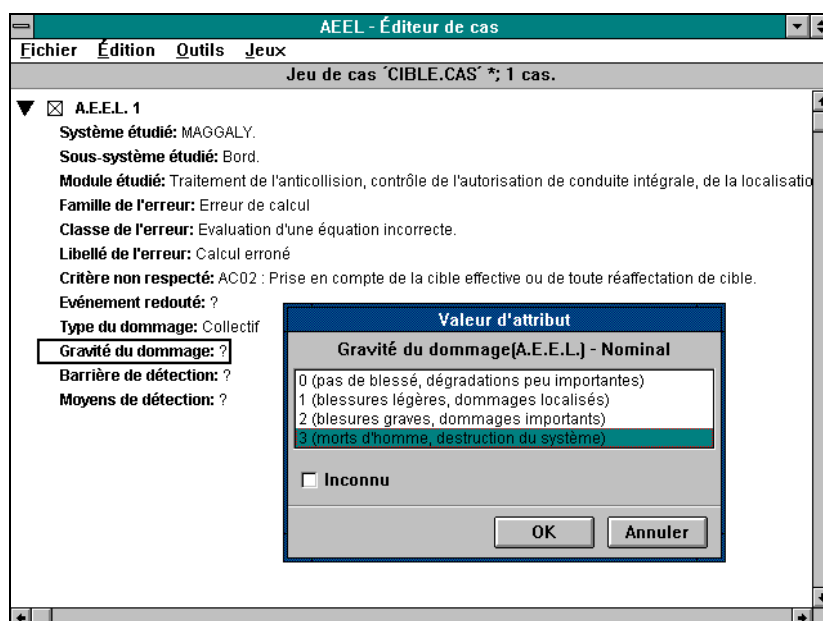
**Figure 14.** Choice of the descriptor order in the indexing tree.



**Figure 15.** Setting up the matching strategy (similarity measure).

### 7.5. 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. Figure 16 shows an example of case entry. We note that the attributes whose value is “?” have not been informed yet and are unknown by default. In addition, we will leave the concept descriptor “dreaded event” unknown because it represents the solution we are looking for in the source case base.



**Figure 16.** Example of the input of a target case.

### 7.6. 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 [53] 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 [54].

From:

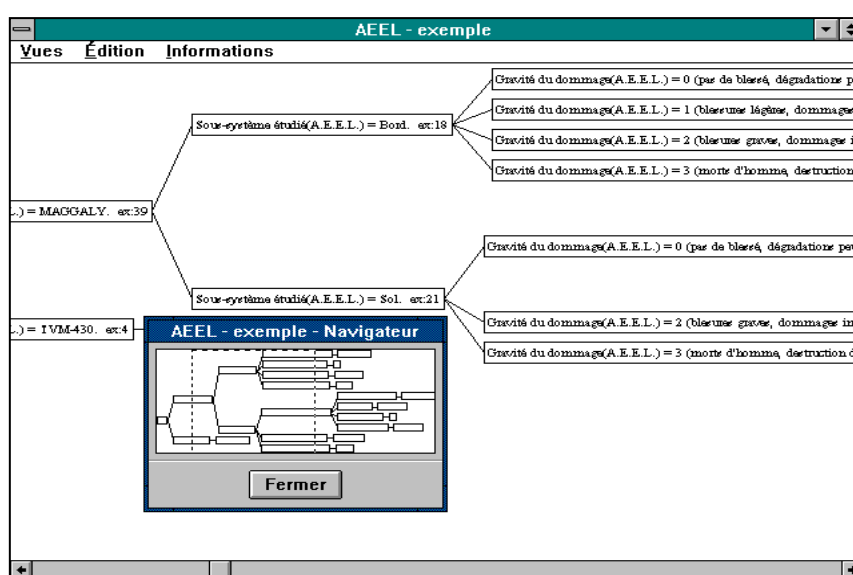
- A set of exclusive classes  $\{C_1, C_2, \dots, C_k\}$ ;
- A set of examples  $\{E_1, E_2, \dots, E_n\}$  represented in the form of pairs (attribute/value) and partitioned in classes  $C_i$ ;

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 17), 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”.

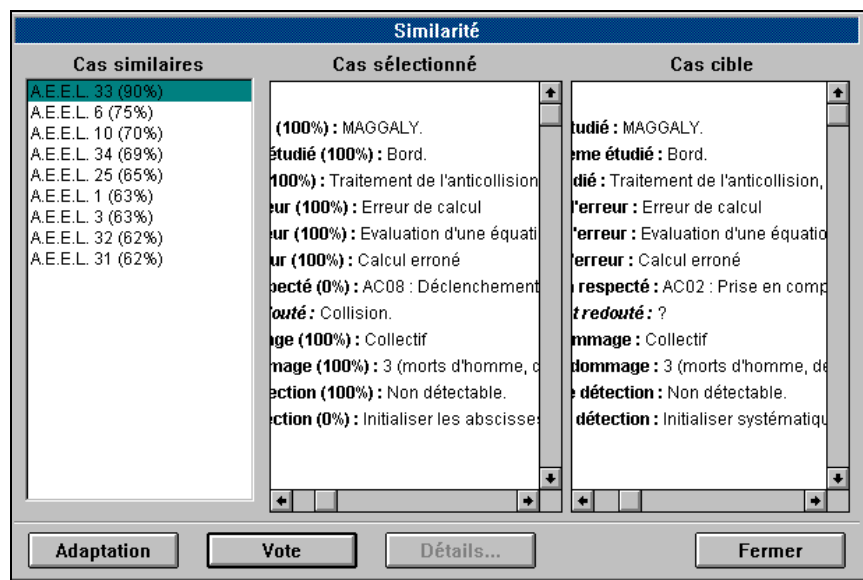


**Figure 17.** Example of the instances base hierarchy.

### 7.7. 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 similar to the SEEA cases to be evaluated (target case) and who share the most important characteristics. The screen shown in Figure 18 shows, for our example, the result of ReCall's 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).

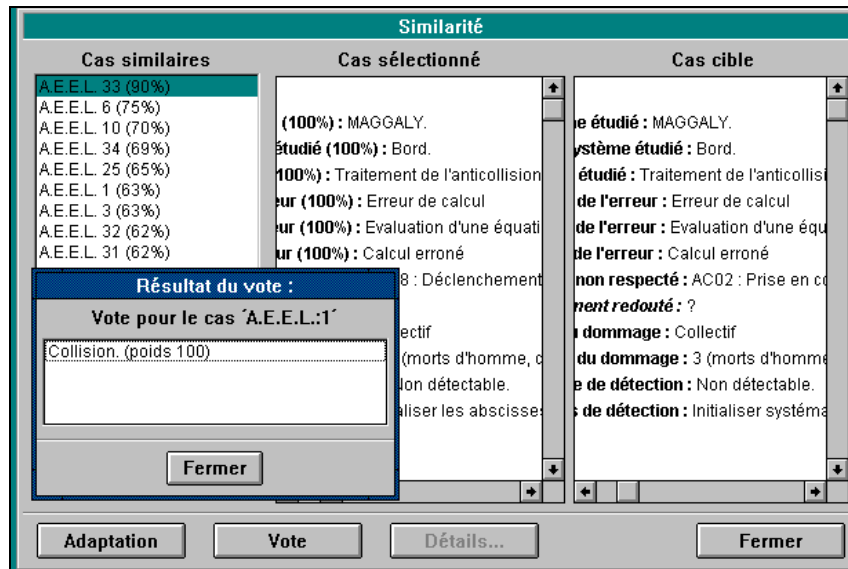


**Figure 18.** Visualization of similar cases extracted from the case base.

### 7.8. 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 19, 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.



**Figure 19.** Example of the reference cases consultation and the vote technique use.

**Table 2.** Example of a solution proposed to solve a new problem from historical knowledge.

Problem		Solution	
Target case (Scenario devised by the domain expert)		Solution to the problem (Potential event proposed by the tool)	
ATTRIBUTE	VALUE	ATTRIBUTE	VALUE
Studied system:	Maggaly-line D		
Studied subsystem	Bor of the train		
Studied module	Anti-collision treatment		
Family of error	Calculation error		
Class of the error	Evaluation of an incorrect equation		
Wording of the error	Erroneous calculation		
Criterion not respected	AC02 (taking into account of the effective target)		
Type of damage	collective		
Seriousness of damage	Level 3		
Detection barrier	Undetectable		
Detection means	Systematically initialize		
		Dreaded event	Collision



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).

### *7.9. 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 relaunch 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 has not yet completed.

## **8. Discussion**

Examination of the safety of a guided or fully automated rail transportation system (without a driver on board the train) requires the use of several techniques, methods, procedures, standards and regulations to ensure that the system, hardware, human-machine interfaces, the environment and human operators do not in any way present a particular risk that can lead to dramatic consequences and intolerable damage to humans, the system and its environment. 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. To this end, we have shown in this article that the success of a knowledge-based system depends essentially on this joint use of machine learning and knowledge acquisition. 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).

With regard to so-called “noisy” data, this is one of the important steps in the process of extracting, structuring and representing the data needed to study the problem. Real-world data tends to be incomplete, noisy, and inconsistent. When preprocessing the data and with the contribution of the domain expert, it is necessary to correctly inform the missing values, to attenuate the noise and to correct any inconsistencies. The quality and capacity of a learning system depends to a large extent on the consistency and completeness of the data to be processed. There are several sources of “noisy data”: The noise can result from the formalization of the initial theory of the domain by the expert (absence or ambiguity of the descriptors), be intrinsic to the domain (uncertainty of a measurement, for example) or resulting from errors when entering the description of the learning examples. Noise detection and processing can occur at many levels of the knowledge acquisition process. One can act during the collection of examples to reinforce the theory of the field when it proves to be insufficient. One can also exploit the disagreements detected between the descriptions of the same examples provided by different experts, in order to obtain information on the nature of the noise. A particular noise processing problem concerns the missing information in the description of the learning examples and more precisely the “missing attribute value”. There are several ways to deal with missing attribute values: 1) The missing value is considered arbitrary or indifferent and can take any possible values, 2) the valueless attribute is considered by the domain expert to be meaningless or not meaningful, 3) the Attributes without values receive the value of the most common attribute in the example class, as is the case in the ID3 algorithm, 4) the incompleteness of the examples is managed by taking into account an explicit knowledge of the domain and described in the form of rules (or axioms) and 5) in other approaches, the incompleteness of the data is palliated by a probabilistic evaluation of the missing values.

In general, we think that the method of treating an incomplete example with a hypothetical value can alter the coherence of the learning examples and lead to semantically unjustified information from the start. This is all the more important if it is a relevant attribute for the description of the example considered. Indeed, in some practical cases, the domain expert may voluntarily not assign a value to an attribute if he considers it irrelevant to discriminate against an example of his counterexamples or if he considers that his judgment is too imprecise to be issued. Finally, an omission on his part when describing an example cannot be ruled out. As part of our application, we recommend the solution proposed by QUINLIN [53] in the ID3 algorithm: the missing value receives the value of the most common attribute in the example class (case).

## 9. Conclusion

This article has presented a contribution to the improvement of conventional methods used to analyze and evaluate the safety of automatic devices in guided rail transport systems. 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. In addition, the SEEA knowledge representation model provides rail safety experts with a methodological framework for better structuring and conceptualizing their knowledge and skills. The study has two main objectives: firstly to record and store experience concerning safety analyses, and secondly to assist those involved in the development and assessment of the systems in the demanding task of evaluating safety studies and in particular the method of SEEA. 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 SEEA 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 the demonstration model.

## Conflict of interest

The author declares that there is no conflict of interest in this paper.

## References

1. CENELEC-EN 50129 (2003) Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling, 1 February 2003, 98.
2. Aamodt Norme AFNOR (1990) Installations fixes et matériel roulant ferroviaires. Informatique – Sécurité de fonctionnement des logiciels, Norme française F 71012 et F 71 013, 1990.
3. Thireau P (1986) Méthodologie d'Analyse des Effets des Erreurs du Logiciel (AEEL) appliqué à l'étude d'un logiciel de haute sécurité 5<sup>e</sup> colloque international de fiabilité et de maintenabilité Biarritz, France, 1986.
4. Hadj-Mabrouk H (2007) 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, Volume 2, chapitre 4, Éditions Hermes/Lavoisier, 123–148.
5. Mabrouk HH (2017) Machine learning from experience feedback on accidents in transport. 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 246–251.

6. Hadj-Mabrouk H (2017) Contribution of learning Charade system of rules for the prevention of rail accidents. *Intell Decis Technol* 11: 477–485.
7. Hadj-Mabrouk H, (2018) 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*, 383–394.
8. Aussenac G, Gandon F (2013) From the knowledge acquisition bottleneck to the knowledge acquisition overflow: A brief French history of knowledge acquisition. *Int J Hum-Comput St* 71: 157–165.
9. Gaines BR (2012) Knowledge acquisition: Past, present, and future. *Int J Hum-Comput St* 71: 135–156.
10. Dieng R (1990) Méthodes et outils d'acquisition des connaissances. ERGO IA90, Biarritz, France, 19 à 21 septembre.
11. Kodratoff Y (1986) Leçons d'apprentissage symbolique automatique. Cepadues éd., Toulouse, France.
12. Ganascia JG (2007) *L'intelligence artificielle*. Cavalier Bleu Eds. ISBN: 978-2-84670-165-5, 128.
13. Ganascia JG (2011) Logical Induction, Machine Learning and Human Creativity. Switching Codes, University of Chicago Press, ISBN: 978022603830, 2011.
14. Michalski RS, Wojtusiak J (2012) Reasoning with missing, not-applicable and irrelevant meta-values in concept learning and pattern discovery. *J Intell Inf Syst* 39: 141–166.
15. Jamal S, Goyal S, Grover A, et al. (2018) Machine Learning: What, Why, and How? In: Shanker A (Eds.), *Bioinformatics: Sequences, Structures, Phylogeny*, Springer, Singapore, 359–374.
16. Aamodt A, Plaza E (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun* 7: 39–52.
17. Harmon P (1991) Case-based reasoning II. *Intelligent Software Strategies*, 7: 1–9.
18. Kolodner J (1992) An introduction to case-based reasoning. *Artif Intell Rev* 6: 3–34.
19. Kolodner J (1993) Case-Based Reasoning. Morgan-Kaufmann Pub. Inc., 668.
20. Leake D, (1996) CBR in Context: The present and future, In: Leake D (ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/MIT Press, 1–30.
21. Mott S (1993) Case-based reasoning: Market, applications, and fit with other technologies. *Expert Syst Appl* 6: 97–104.
22. Pinson S, Demourieux M, Laasri B, et al. (1993) Le Raisonnement à Partir de Cas: panorama et modélisation dynamique. Séminaire CBR, LAFORIA, Rapport 93/42, 1er octobre 1993.
23. Slade S (1991) Case-based reasoning: A research paradigm. *AI Mag* 12: 42–55.
24. Bergmeir C, Sáinz G, Bertrand CM, et al. (2013) 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.
25. Fay A (2000) A fuzzy knowledge-based system for railway traffic control. *Eng Appl Artif Intel* 13: 719–729.
26. Santur Y, Karaköse M, Akin E (2017) A new rail inspection method based on deep learning using laser cameras. International Artificial Intelligence and Data Processing Symposium (IDAP), 16–17 Sept. 2017.

27. Faghieh-Roohi S, Hajizadeh S, Núñez A, et al. (2016) Deep convolutional neural networks for detection of rail surface defects. International Joint Conference on Neural Networks (IJCNN), 24–29 July 2016, Canada.
28. Ghofrانيا F, He Q, Goverde R, et al. (2018) Recent applications of big data analytics in railway transportation systems: A survey. *Transport Res C-Emer* 90: 226–246.
29. Thaduri A, Galar D, Kumar U (2015) Railway assets: A potential domain for big data analytics. *Procedia Comput Sci* 53: 457–467.
30. Attoh-Okine N (2014) Big data challenges in railway engineering. IEEE International Conference on Big Data (Big Data), 27–30 Oct. 2014, Washington, DC, USA.
31. Peter Hughes (2018) Making the railway safer with big data. Available from: <http://www.railtechnologymagazine.com/Comment/making-the-railway-safer-with-big-data>.
32. Vicki Hayward (2018) Big data & the Digital Railway. Available from: <https://on-trac.co.uk/big-data-digital-railway/>.
33. Bernard Marr (2017) 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>.
34. Zubair M, Khan MJ, Awais M (2012) Prediction and analysis of air incidents and accidents using case-based reasoning. Third Global Congress on Intelligent Systems, 6–8 Nov. 2012, Wuhan, China.
35. Khattak A, Kanafani A (1996) Case-based reasoning: A planning tool for intelligent transportation systems. *Transport Res C-Emer* 4: 267–288.
36. Sadeka A, Smith B, Demetsky M (2001) A prototype case-based reasoning system for real-time freeway traffic routing. *Transport Res C-Emer* 9: 353–380.
37. Sadek A, Demetsky M, Smith B (2002) Case-Based Reasoning for Real-Time Traffic Flow Management. *Comput-Aided Civ Inf*.
38. Zhenlong L, Xiaohua Z (2008) A case-based reasoning approach to urban intersection control. 7th World Congress on Intelligent Control and Automation, 25–27 June 2008, Chongqing, China.
39. Li K, Waters NM, (2005) Transportation Networks, Case-Based Reasoning and Traffic Collision Analysis: A Methodology for the 21<sup>st</sup> Century, In: Reggiani A, Schintler LA (eds.), *Methods and Models in Transport and Telecommunications*, Advances in Spatial Science. Springer, Berlin, Heidelberg, 63–92.
40. Kofod-Petersen A, Andersen OJ, Aamodt A, (2014) 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.
41. Louati A, Elkosantini S, Darmoul S, et al. (2016) A case-based reasoning system to control traffic at signalized intersections. *IFAC-Papers On Line* 49: 149–154.
42. Begum S, Ahmed MU, Funk P, et al. (2012) 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. 2012, Wroclaw, Poland.

43. Zhong Q, Zhang G, (2017) 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.
44. Varma A, Roddy N (1999) ICARUS: Design and deployment of a case-based reasoning system for locomotive diagnostics. *Eng Appl Artif Intel* 12: 681–690.
45. Johnson C (2000) 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.
46. Cui Y, Tang Z, Dai H (2005) 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 June 2005, Chongqing, China.
47. Li X, Yu K (2010) 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. 2010, Dalian, China.
48. Lu Y, Li Q, Xiao W (2013) Case-based reasoning for automated safety risk analysis on subway operation: Case representation and retrieval. *Safety Sci* 57: 75–81.
49. de Souza VDM, Borges AP, Sato DMV, et al. (2016) 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.
50. Zhao H, Chen H, Dong W, et al. (2017) 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.
51. Darricau M (1995) 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, juin 1995.
52. Darricau M, Hadj-Mabrouk H (1996) 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.
53. Quinlan JR (1986) Induction of Decision Trees. *Mach Learn* 1: 81–106.
54. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27: 379–423.



AIMS Press

© 2019 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)