In: *Complex Systems Design & Management, Proceedings of the Fifth International Conference on Complex Systems Design & Management CSD&M 2014*, Springer, pp. 163-174, 2015

How can usage monitoring improve resilience?

Jean-René Ruault[*],[**], Frédéric Vanderhaegen[*], Christophe Kolski[*]

* Univ Lille Nord de France, 59000 Lille, France

UVHC, LAMIH, 59313 Valenciennes, France

CNRS, UMR 8201, 59313 Valenciennes, France

(surname.name@univ-valenciennes.fr)

**DGA, 7-9 rue des Mathurins, F-92221 Bagneux, France

(jean-rene.ruault@intradef.gouv.fr)

Abstract

Resilience and systems engineering are key issues for critical systems. The operational usage and states of such systems are quite different from reference ones, generating drift and generate risks. This article suggests functional and physical architectures that fit resilience. Four functions relate to resilience (avoidance, resistance, recovery, adaptation). We develop the avoidance one and define a usage monitoring system that implements it. The case study concerns a railway accident that occurred at Aldershot, Canada. We explain the origin of the gap leading to the accident. The usage monitoring system would allow human operators to understand the situation and avoid the accident.

Introduction

Nowadays, resilience is a key issue for complex system, with many books and articles dealing with resilience [3], [10], [12], [13], whatever the domain, in order to cope with unexpected events. Systems engineering is another key issue [8], [6]. Many critical and complex systems show a very long lifecycle. We can't foresee all the operational situations that they will meet. The resilience of a system facing unforeseeable events is a critical challenge. The paper suggests a solution to monitor system real states in order to assess drift and to alert human operator of the proximity of hazard. The first part of this paper summarizes the state of the art, for systems engineering and resilience. The second one details a design pattern fit to resilient systems. It contents functional and physical architecture models. It details impacts on the usage monitoring components and on the user interface. The third part applies these concepts to a case study, in the railway domain.

State of the art

The state of the art details the main concepts upon which is based this paper, that are systems engineering, systems architecture, systems modeling language, as well as resilience.

Systems engineering, architecture, SysML

The ISO 15288 standard defines a system as “a combination of interacting elements organized to achieve one or more stated purposes” [6], while IEEE 1220 defines it as “a set or arrangement of elements [people, products (hardware and software) and processes (facilities, equipment, material, and procedures)] that are related, and whose behavior satisfies operational needs and provides for the life cycle sustainment of the products” [5].

For instance, a railway system allows transporting travelers and fret from point to point. Such a system encompasses: end products or services, for instance, travelers transportation, fret transportation; equipments and devices producing these end products and services that are trains, stations, traffic management systems, sale systems; enabling systems, that are CASE tools or test tools; end users specified processes and activities, such as driving train, managing traffic, vending tickets, conforming regulation rules; end users specified profiles, roles and responsibilities, such as engineers, traffic managers, structure organization, that specifies both end users profiles, roles and responsibilities, as well as processes and activities; resources, such as electricity.

The “system in use” is quite different to the “system as designed”. End users behave to reach performance goals, control their activities, adapting them function of contextual contingencies and improvement of performance requirements, and resolve system dysfunctions and failures. Most of the security and safety analysis are based upon foreseen and predictable failures and do not take into account unforeseen and unpredictable events. So, the system is not designed to perform in such a way and the users have to resolve the gap between the unpredictable events and the system functions. The system architecture models describe the organization of the functions and the components of the system. Main architectural points of view are:

- The operational one: why is the system designed and built? What are its missions and goals? What are the operational missions in which it will be used?
- The functional one: what are the services that the system provides to its environment? What is the organization of these services?
- The physical one: how the system’s components interact together in order to provide these services?

Nowadays, the system modeling language (SysML) provides a set of diagrams in order to elaborate these models [1]. These diagrams allow modeling structure and behavior of a system. Moreover, including requirements diagram, SysML allows traceability between models and requirements. SysML is a key driver of model based systems engineering. This set of diagrams contents: (1) Structure diagrams set (block definition diagram and internal block diagram); (2) Behavior diagrams set (activity diagram, sequence diagram, state machine diagram and use case diagram); (3) Parametric diagram; (4) Requirements diagram and, (5) Package diagram.

Resilience functions

Resilience is an extrinsic relational property of a system. It characterizes its property "to cope" with adversity where the disturbance is unforeseeable. Authors of the "Resilience engineering" book [3] states: "Resilience is capacity of a system or an organization to react and recover after a disturbance, with a minimal effect on dynamic stability". Moreover, resilience is complementary and adds value to other system safety method.

Luzeaux [8] characterizes resilience as a "management at the border of the domain of application… The challenges linked to resilience include the management of that which is uncertain or unplanned, accidents, the transition between more or less catastrophic circumstances while avoiding a true catastrophe, and the return to a more normal operational status". Luzeaux [8] differentiates four main resilience functions which complements each other: 1) avoidance (capacity for anticipation); 2) resistance (capacity for absorption); 3) adaptation (capacity for reconfiguration), and 4) recovery (capacity for restoration). We focus now on the first key function for resilience: avoidance [8]. The avoidance function consists of acquiring information at the operators' level in order to anticipate and to avoid the accident, that is: (1) To obtain a representation of the environment; (2) To obtain a representation of the system dynamics: (3) To identify the environment states that were not envisioned; (4) To evaluate the instantaneous or trend drifts; (5) To evaluate the proximity of the state of the system compared to the hazard.

These functional elements impact the architecture of the system of interest, enabling systems architecture, since we give to operators an appropriate situation awareness representation. Resilience is the dynamic process that allows the crew to understand the current situation, to learn and develop adequate behaviors to take into account environment adversities and to adapt as well as possible. It is the capacity of a sociotechnical system to continue to evolve and fulfill its operational mission in spite of the difficult conditions, serious constraints or events, sometimes severe damages or losses. This adjustment capability is based upon the dynamic process of "visual piloting". The system must have a great capacity to estimate its position with regard to the danger zone [8]. The system must be designed to cope with uncertainty. It is necessary to specify the envelope of required, desirable, even acceptable, execution and to require that the system recognizes the situations where it is likely to leave this envelope. "Resilience is obtained via the system capability to monitor conditions at the edges of the performance envelope, determining their value and the usual distance from the edge and the ability to adapt the operational behavior of the system to potential developments in the envelope…" [8]. The objective is to qualify and quantify the drift of the system towards the state of failure before a major breakdown occurs.

In many cases, the system has been designed to be safe under specified conditions, but there are no means to monitor the system when it operates under unspecified conditions, and to reassess actual risk. Safety under this situation is neither monitored nor controlled. The resilient management consists of clear, relevant and shared situation awareness, among all the communities, which implies to assess the gap between the specified path and the actual one as usual fluctuations or, on the opposite, the trend of a forecast latent deviation.

Hardy [2] expresses that "plans that do not reflect reality may create the impression that effective organization is in place and that risks have been reduced, when in fact large risks could exist". This difference may grow from the beginning of the operation of the system, from step to step, generating a gap between the plans and the reality. The figure 1 expresses this difference and the gap. The specified path deals with the specified task [7], or the work-as-imagined, taking place along the time. It contains specified local variability included within tolerance margins that is everyday or 'normal' variability [4] as defined *a priori*.

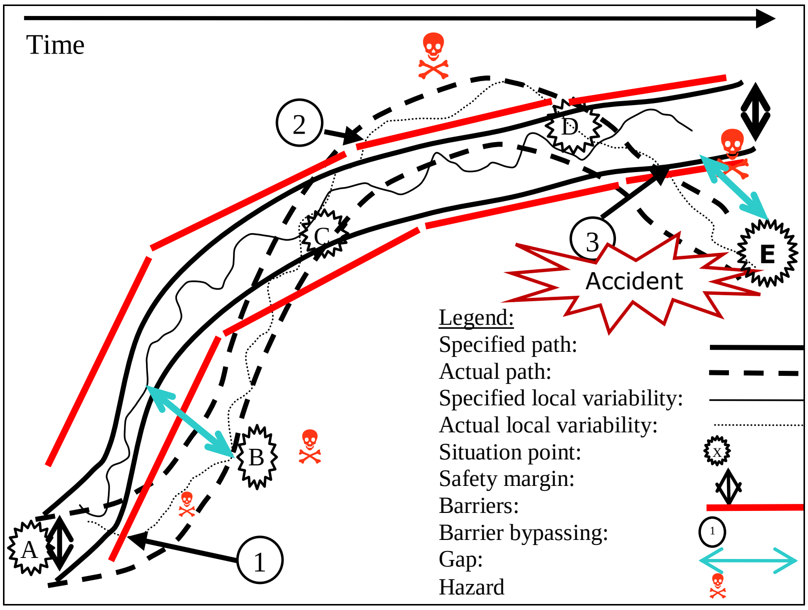

Fig. 1 Specified and actual paths of a sociotechnical system [10].

The actual path, among other possible ones, denotes the actual activity [7], or the work-as-done [4], of the sociotechnical systems, function of met contingencies. This actual path contains actual local variability, since these contingencies are not stable and linear. The gap between these two paths is due to unusual conditions, 'out-of-range' variability [4], that is not an isolated case, but a huge trend. These unusual conditions may be new and unforeseeable working environments conditions. By coming of A, the real dynamics by-passes the barrier in 1 (cf. figure 1), moves towards B, then C, to join D and E by by-passing of new the barrier in 2 and 3. This real dynamics of A in E expresses a gap which can be far from the prescribed dynamics, as it is the case B. Nobody can estimate this gap. Nobody is conscious of situation and can estimate the risk infers by the drift. Stage after stage, the dangerous actions increase the risks (☠), until the accident (E). The critical issue is the capacity of the user interface to give to the operators a shared situation awareness and to allow a navigation at sight. It is necessary to compare the real states and usage of the system and the reference ones.

Design pattern fit to resilient systems

We elaborate a design pattern that fits resilient systems. This design pattern declines the avoidance function in a functional architecture, then in a physical one. We will decline the other functions in further articles. The main functions of the resilience are avoidance, resistance, recovery and adaption. In this article, we detail the avoidance function.

Functional architecture: monitor system's usage and current state

The goal of this function is to be aware of the current situation compared with the specified one that is the drift compared with the nominal path, the proximity of hazard, and the safety margins.

It consists of gathering information about the system, its dynamics, its environment, and alerting operator when the system deviates from its nominal path. This function (figure 2) is decomposed into these following four functions on which we focus: (1) To obtain a representation of the system dynamics; (2) To evaluate drifts; (3) To evaluate proximity of hazard and (4) To alert operators. These functions are allocated to components in the physical architecture of the usage monitoring system.

Physical architecture: usage monitoring components

These four functions are allocated to the usage monitoring system. It provides two sets of services respectively for these two functions: (1) To obtain a representation of the system dynamics; to gather usage from operating parts of a system; (2) To alert operators; to express warning.

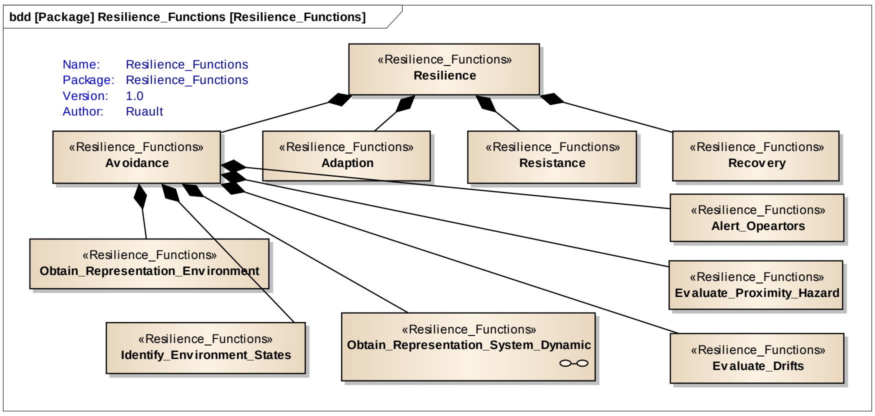
bdd [Package] Resilience_Functions [Resilience_Functions]
Name: Resilience_Functions
Package: Resilience_Functions
Version: 1.0
Author: Ruault
«Resilience_Functions» Resilience
«Resilience_Functions» Avoidance
«Resilience_Functions» Adaption
«Resilience_Functions» Resistance
«Resilience_Functions» Recovery
«Resilience_Functions» Alert_Opeartors
«Resilience_Functions» Evaluate_Proximity_Hazard
«Resilience_Functions» Evaluate_Drifts
«Resilience_Functions» Obtain_Representation_Environment
«Resilience_Functions» Identify_Environment_States
«Resilience_Functions» Obtain_Representation_System_Dynamic

Fig. 2 Functional decomposition of the avoidance function (block definition diagram).

The two other functions are implemented inside the usage monitoring system that encompasses a set of components that are:

- Usage sensor proxies are closely nested to the security devices or other components of the systems, gather their states and usages and send them to the respective usage sensors. Each component and security device that contributes to resilience has a usage sensor proxy that fits to it.
- Usage sensors get the states and usages of devices and components and translate them in order to be analyzed. Each component and security device that contributes to resilience has a usage sensor that fits to it.
- Current state repository stores the data coming from the usage sensors whatever they are, in order to assess trend drifts. It deals with reality [2].
- Reference state repository contains models that specify security, including specified variability, barriers characteristics, as well as more specific data. It deals with plan [2].
- States comparison engine compares the current states and the reference ones, in order to assess drifts and evaluate the proximity of hazard. It sends warning levels, safety margins and drifts to the user interface proxy.
- User interface proxy is closely nested to the other user interface of the system and expresses warning in order to alert human operators.

The table 1 shows the allocation of resilience functions on the usage monitoring system components.

Table 1: Allocation of resilience functions on the usage monitoring system components.

Functions	Obtain a representation of the system dynamics	Evaluate the drifts	Evaluate the proximity of the system compared with the hazard	Alert operators
Components	Public service	Internal function	Internal function	Public service
Usage sensor proxy	X			
Usage sensor	X			
Current state repository		X	X	
Reference state repository		X	X	
States comparison engine		X	X	
User interface proxy				X

The block definition diagram (figure 3) shows the physical architecture of the usage monitoring system, and each component with its operations and attributes, that have to be tailored in order to fit domain specificities and safety stakes.

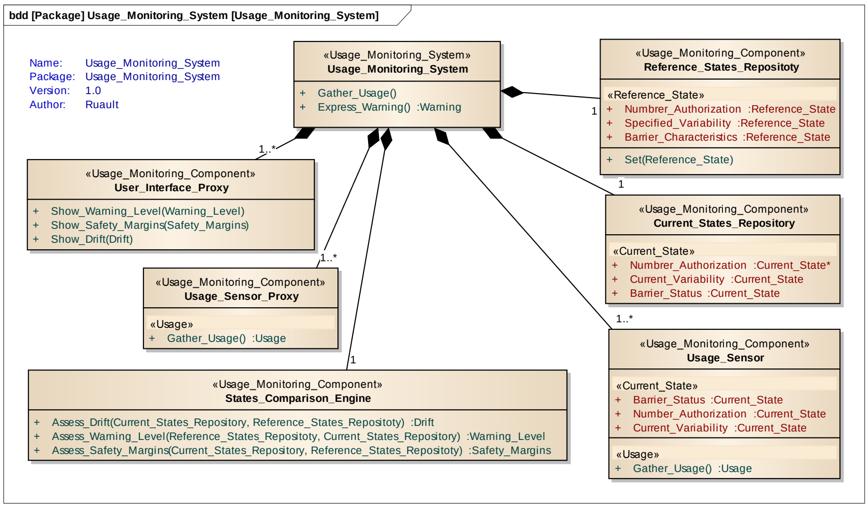
bdd [Package] Usage_Monitoring_System [Usage_Monitoring_System]
Name: Usage_Monitoring_System
Package: Usage_Monitoring_System
Version: 1.0
Author: Ruault
«Usage_Monitoring_System»
Usage_Monitoring_System
+ Gather_Usage()
+ Express_Warning() :Warning
«Usage_Monitoring_Component»
Reference_States_Repositoty
«Reference_State»
+ Numbrer_Authorization :Reference_State
+ Specified_Variability :Reference_State
+ Barrier_Characteristics :Reference_State
+ Set(Reference_State)
1
1..*
«Usage_Monitoring_Component»
User_Interface_Proxy
+ Show_Warning_Level(Warning_Level)
+ Show_Safety_Margins(Safety_Margins)
+ Show_Drift(Drift)
1
«Usage_Monitoring_Component»
Current_States_Repository
«Current_State»
+ Numbrer_Authorization :Current_State*
+ Current_Variability :Current_State
+ Barrier_Status :Current_State
1..*
«Usage_Monitoring_Component»
Usage_Sensor_Proxy
«Usage»
+ Gather_Usage() :Usage
1..*
1
«Usage_Monitoring_Component»
Usage_Sensor
«Current_State»
+ Barrier_Status :Current_State
+ Number_Authorization :Current_State
+ Current_Variability :Current_State
«Usage»
+ Gather_Usage() :Usage
«Usage_Monitoring_Component»
States_Comparison_Engine
+ Assess_Drift(Current_States_Repository, Reference_States_Repositoty) :Drift
+ Assess_Warning_Level(Reference_States_Repositoty, Current_States_Repository) :Warning_Level
+ Assess_Safety_Margins(Current_States_Repository, Reference_States_Repositoty) :Safety_Margins

Fig. 3 Physical architecture of the usage monitoring system.

Impacts of the usage monitoring components upon functional and physical architectures of whole system

The system architecture must evolve in order to link together the operating system and the usage monitoring system. We differentiate two parts. On the one hand, the operating system implements the core functionalities, that is the part that reach the goals and realizes the operational missions. On the other hand, the usage monitoring system implements the avoidance functions of the resilience. Any kind of systems can implement this pattern, rail traffic management systems as well as trains, tracks or stations (figure 4).

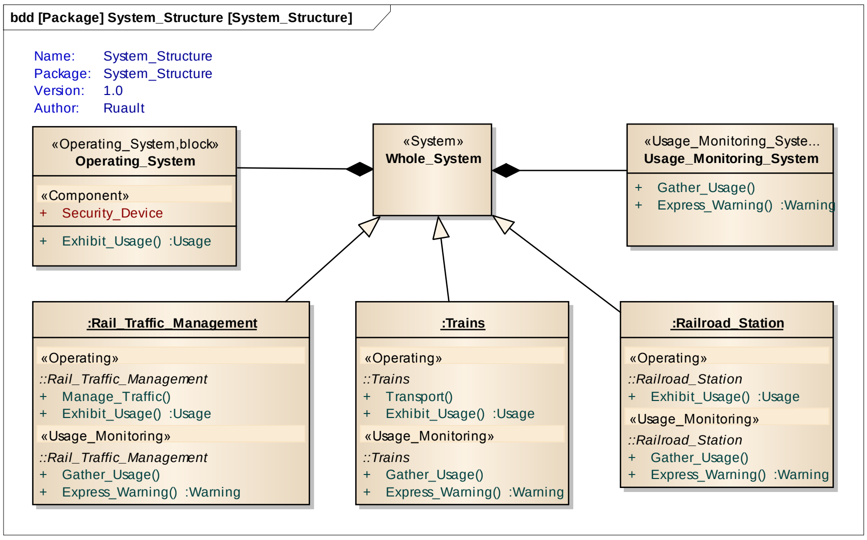

Fig. 4 A whole system containing an operating system part and an usage monitoring system one.

For each of these different systems, their operating parts exhibit usage and state, as well as their specific functionalities. On the other side, their usage monitoring parts gather usage and express warning. That implies interfaces and flows between these two types of parts (figure 5). The operating system exhibits usage, for instance current variability, barrier state, among other safety information. The usage monitoring system gathers these information and, function of the real state of the operating system, expresses warning in order to alert the operators, such as safety margins, drift or proximity of hazard.

Impacts of the usage monitoring components upon user interface

The system architecture must evolve in order to express the warning to the operators, via the relevant user interfaces. This relies on the capacity to measure its current internal states and explain the gap between them and the reference ones. These interfaces must be designed in order to express safety margins, hazard proximity and increase of risk level.

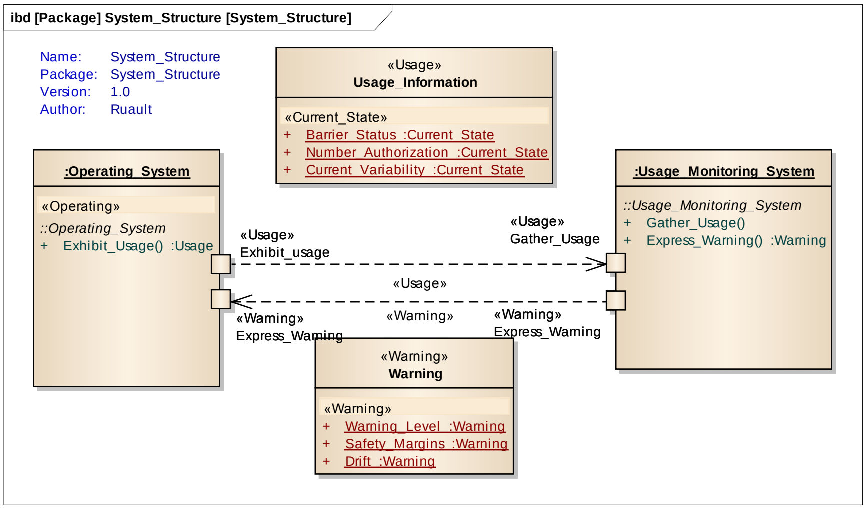

Fig. 5 Interfaces and flows between operating system and usage monitoring system (internal block diagram).

So, the operators can regulate their activities, evaluate the differences between the current system situation and the system field of definition, and detect, as soon as possible, the migration or compensation mechanisms. We suggest a solution that needs to be assessed with operators. It expresses the progressive drift from secure situation to risky one [10]. The operators must be able to see that the system is in a high risk zone with catastrophic consequences. Since that the operators are awarded of the proximity of hazard, they can take care, improve procedures and monitor the real state of the system.

Case study: railway accident

The application is the railway accident that occurred at Aldershot station [11].

According to the accident report [11], "On 26 February 2012, VIA Rail Canada Inc. passenger train No. 92 (VIA 92) was preceding eastward from Niagara Falls to Toronto, Ontario, on track 2 of the Canadian National Oakville Subdivision near Burlington, Ontario. VIA 92". The investigations show that while approaching the Aldershot station, the crew encountered a first signal (Clear to Limited) and then another one (Clear to Slow). The signals were specifying to proceed and approach signal 334T2 located east of the Aldershot station at 24 km/h (15 mph). This consecutive information was part of the signal indications governing VIA's 92 movement in order to pass from the track 2 to the track 3. However, the stop at the Aldershot station was an event that interrupted the signal progression. Hence the crew was more preoccupied by stopping the train at the station than proceeding to the signal 334T2 with appropriate speed. During the stop, there was no further indication to remind the crew of the previous signal. This event laid to the interruption of the signal indications, promoting oblivion of the past information. Moreover, in 99 % of the cases, the train of the company circulated on the track 2. This day, works were realized on the track 1 and the track 2. An authorization to occupy the track was granted to the team in charge of these works by the controller of the rail traffic. The team of exploitation of the train was not informed about these works. The train had to pass of the track 2 in the track 3, via a crossover between track 2 and 3. This instruction was communicated by the railway signals which the team of exploitation is supposed to apply scrupulously. The speed on this connection was limited to 24 km/h (15 mph). The team of exploitation understood too late the situation. The train entered on the connection a 108 km/h speed (67 mph) and went leaned. When a passage of the track 2 to the track 3 is necessary, this passage is realized on a crossover for which the authorized maximal speed is 72.42 km/h (45 mph). The day of the accident, the situation was quite different. The drivers VIA Rail had to pass of the track 2 in the track 3 on a crossover as which the authorized maximal speed was of 24 km/h (15 mph). This situation, the day of the accident, generated an important gap between the specified speed, adapted to pass on the crossover, and the real speed of the train. The instruction of speed was shown on the railway signals, before Aldershot station in which the train stopped. There was no reminder of the specified speed when the train restarted of Aldershot station. The speed of the train was excessive and the capacity to brake insufficient to enter on the points limited in 24 km/h (15 mph). The drivers were not conscious of this gap. When they understood the situation, it was too late. The drivers were not able to avoid the accident.

An usage monitoring system is useful for detecting the major violations of safety. In the case study of the Aldershot station railway accident, a usage monitoring system would have been helpful for the following issues:

- Detecting the speed excess by comparing it to the accepted threshold and presenting the evidence of a speed exceeding to the operators.
- Managing the rail crossing between rail 2 to rail 3 by alerting the operators or presenting a visual device with the maneuver to be accomplished.
- Reporting the railway signals to the operators in real-time in order to keep them informed and secures the situation awareness of the operating crew.

Indeed, from the moment when the VIA 92 entered the crossover No. 5 (1) with excessive speed, consistent with the crew misperception of the railway signals (2), it could be imagined that the usage monitoring system would have reported the anomaly through the usage sensor components. Hence, during the stop at the Aldershot station, interrupting the continuous progression of signals (3) (4) the usage monitoring system might have been helpful to the operators for understanding the situation compared to the reference state (specified path cf. Fig 1) with the state comparison engine component.

Among other possibilities, we suggest an architecture allowing showing to the drivers this gap so that they become aware of the situation and adapt the speed in a appropriate way. The specified speed and speed limited to 24 km/h (15 mph) must be transmitted by the traffic management system in the train, in order to the usage monitoring system can compare these speeds with the real one. The drivers owe informed about the speed limited to 24 km/h (15 mph) of the points which they have to take. That needs to communicate to the drivers of the specified speed, the maximal speed authorized on the points, the gap between the specified speed and the real speed, as well as the necessary distance to brake before to engage on the points.

Both rail traffic management system and train system (figure 6) contents an operating subsystem and a usage monitoring one. The train system gathers specified speed and crossover maximum speed in order to evaluate this information compared to the actual speed and alerts the drivers about the over speed.

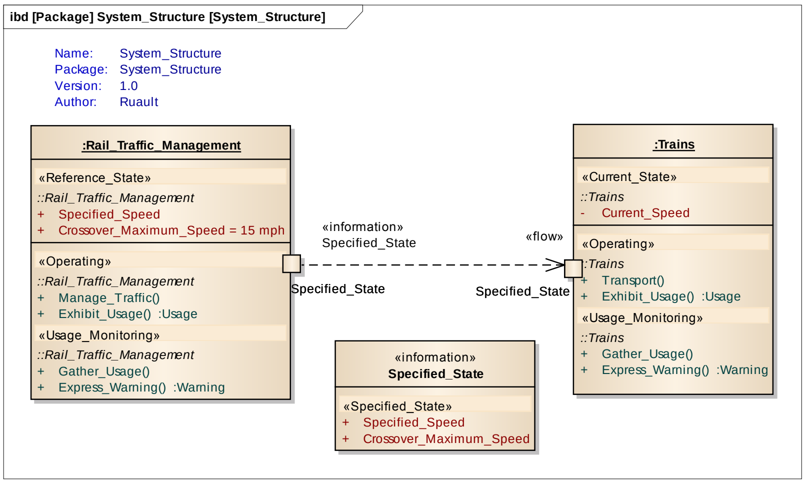

Fig. 6 Communication of specified states from the rail traffic management and the train.

Other solutions exist. For instance, a signal after the station informs all drivers of conditions beyond the station, or an adaptive speed limiting based on signals communicates from train to train, in order to prevent collision.

Conclusion

Our proposal consists in interconnecting the operating system, which realizes the operational missions, and the usage monitoring system. This interconnection allows monitoring the state, the usage of the system, to estimate the gap between the current state of the system and the safe one, to estimate the proximity of hazard, and to inform the operators. The goal is that the operators share a clear, reliable, relevant and updated representation of the operational context as well as the usage of the system, so that they can take the appropriate measures. The first stage, object of this paper, is to design a system architecture implementing this interconnection to monitor the usage of the system. A second stage will consist in widening this architecture to observe the operational context and express it to the operators. A first difficulty lies in the determination of issues to be observed, in particular the behavior which cannot be *a priori* envisioned. A second difficulty lies in the reliability and the fidelity of the measures. Indeed, the lack of information, or false information, would be error prone and accident prone. Finally, benchmark models allowing estimating this drift have to be available in the system in exploitation, and to correspond to the real configuration of the system.

Bibliography

[1] Friedenthal S, Moore A, & Steiner R (2011) A Practical Guide to SysML, Morgan Kaufmann; 2nd edition.

[2] Hardy T-L (2010) The system safety skeptic. Author-House, Bloomington, USA.

[3] Hollnagel E, Woods DD, & Leveson N (2006) Resilience engineering. Concepts and precepts ; Ashgate, Hampshire, Great Britain, 2006

[4] Hollnagel E (2012). FRAM: The Functional Resonance Analysis Method. Ashgate, Hampshire, Great Britain.

[5] IEEE Std 1220 (2005) IEEE Standard for Application and Management of the Systems Engineering Process.

[6] ISO/IEC 15288 (2008) Systems engineering — System life cycle processes.

[7] Leplat J (1985) Erreur humaine, fiabilité humaine dans le travail. Paris, Armand Colin, 197 pages.

[8] D Luzeaux (2011) Engineering Large-scale Complex Systems in D Luzeaux, J-R Ruault & J-L Wippler, Complex Systems and Systems of Systems Engineering, ISTE Ltd and John Wiley & Sons Inc, 2011.

[9] Ruault J-R, Vanderhaegen F, Luzeaux D (2012) Sociotechnical systems resilience. 22nd Annual INCOSE International Symposium, 9-12 July, 2012, Rome.

[10] Ruault J-R, Vanderhaegen F, Kolski C (2013) Sociotechnical systems resilience: a dissonance engineering point of view. 12th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (august, 11-15), IFAC, Las Vegas, USA.

[11] Transportation Safety Board of Canada (2013) Railway Investigation Report R12T0038, Main-track Derailment VIA Rail Canada Inc. Passenger Train No. 92 Mile 33.23, Canadian National Oakville Subdivision Aldershot, Ontario 26 February 2012.

[12] Zieba S, Polet P, Vanderhaegen F & Debernard S (2010) Principles of adjustable autonomy: a framework for resilient human machine cooperation. Cognition, Technology and work, 12 (3), 193-203.

[13] Zieba S, Polet P, Vanderhaegen F (2011). Using adjustable autonomy and human-machine cooperation for the resilience of a human-machine system, Application to a ground robotic system. Information Sciences, 181, 379-397.