Sysml formalization of the disruption management process in european railways

Birgit Jaekel, Paola Pellegrini, Sonia Sobieraj Richard, Joaquin Rodriguez

TECHNISCHE UNIVERSITÄT DRESDEN
"Friedrich List" Faculty of Transport and Traffic Sciences

French Institute of Science and Technology for Transport, Development and Networks
IFSTTAR

A SYSML FORMALIZATION OF THE DISRUPTION MANAGEMENT PROCESS IN EUROPEAN RAILWAY

Birgit Jaekel (TU Dresden) Sonia Sobieraj Richard (IFSTTAR)
Paola Pellegrini (IFSTTAR) Joaquín Rodriguez (IFSTTAR)

Disruption Management in Railway Operation

Disruption management in railway operation describes the (re-)actions that are performed to return the system back to its initial operational state after an unexpected event affected the system lastingly. The challenge of disruption management in railway operation is to get back to regular operation after an unexpected event occurs and to minimize loss and negative impacts on the whole railway system. Disruption management for railway operations includes the consideration of constrains (e.g., connection information, resource dependencies), the determination of optimization criteria (e.g., recovering capacity) as well as the optimization of the process with the coordination of individual measures. This complex process is of great significance as the performance of its stakeholders has direct economic consequences.
The organization of disruption management varies across Europe, and sometimes even within a country, depending on locations and physical layouts. So far, there is only little technical support especially designed for disruption management. Contingency plans (in some cases available for infrastructure) only cover strategies within the scope of one organization.
The processes in disruption management and the workload due to cross-organizational communication would benefit from automated decision support that helps to manage large disruptions in a robust, reliable and simple way. Of course, human operators cannot be substituted, but they need to be supported in disruption management decisions, e.g., by providing a measure for the assessment of disruption handling actions. A well defined process of disruption management will take away pressure from decision-makers and lead to more reliable results.

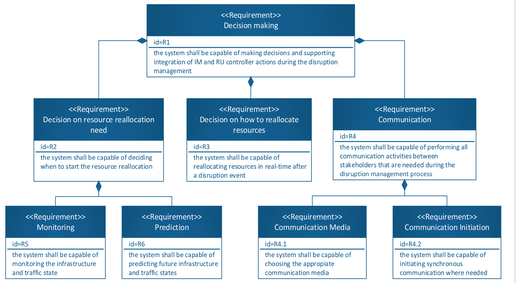

Figure 1: SysML Requirements diagram with main requirements.

Process Requirements

Disruption management relies on several actors who ensure railway operations, as controllers and resource planners from both infrastructure managers and railway undertakings. The variety of the involved actors and their requirements explains why a successful disruption management depends on unambiguous communication between all parties (Figure 2). It is a highly interconnected process with any aspect of decision-making being linked to several other aspects and to the progress achieved within one's own organization and cross-organization. Location, timing, type of incident and severity of incident will all influence the capacity to deliver an alternative timetable, and therefore influence alternative solutions.

System Structure Definition With Block Diagrams

In SysML models, blocks are used to define a system's structure (Figure 2). Block characteristics can be described using properties, operations or references. By compositions and references among blocks, both structural and virtual dependencies can be modeled. Here, block diagrams are used for modelling the disruption management process. They define the relevant properties of the infrastructure, the train definition, the overall schedule, the trips, the events, and the human resources (drivers, conductors and crew). The behavior of the system components is not explicitly modeled here – they are considered black-box systems. This allows the modeling of different systems through the same structure, where blocks simply act as interfaces between possibly different implementations. By doing so, we can represent the disruption management processes implemented in several European countries through a unique model .
Figure 2 shows the model consisting of 3 resource schedules (timetable, rolling stock and crew) and the assets needed to form them. Different arrows indicate the blocks being integral parts of their parents (diamonds) or associated to other blocks (triangle).

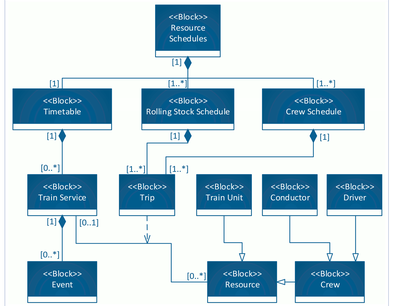

Figure 2: SysML Block diagram.

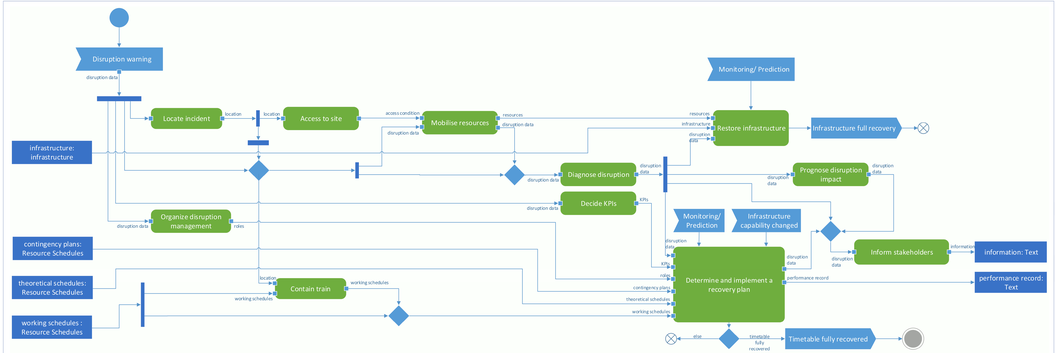

Figure 3: SysML Activity diagram of the disruption management process. Each action's behavior may differ depending on implementation.

Formalization of Disruption Management Processes

In Figure 3, the process starts as soon as a *Disruption warning* signal is received. Four concurrently working actions (*Locate incident*, *Organize disruption management*, *diagnose disruption* and *decide KPI's*) are executed. These four actions invoke activities that refine location, scope, causes of the disruption, roles within the organization and Key Performance Indicators (KPIs) to be used. After having invoked *Locate incident*, the process can proceed with the action *Contain trains* as the location is known and the trains to hold to prevent delay escalation can be determined. The holding of trains is recorded in the timetable of the input parameter working schedules. The changed working schedules are placed on the output flow to be input to *Determine and implement a recovery plan*, to make a decision on the held trains.
The action *Access to site* is invoked simultaneously. It requires the plan, execution or handback of track or electrical isolation to allow access to the site of the incident (e.g., setting signals to red so that track-workers can access the failed point). Once the access conditions are met the action *Mobilise resources* is invoked to move resources (people or plant) to the site of the incident. To do so, the disruption data are also required, which explains the further transmission of this token from the initial fork. When the action *Mobilise resources* is completed, the resources on site can provide new information about the incident. Depending on the nature of the incident, the activities *Access to site*, *Mobilise resources* and the subsequent *Restore Infrastructure* may not imply any actual action. In the *Diagnose Disruption* action, the identified cause of the incident is stored in the corresponding attribute of the output token disruption data. The token is then replicated to enable the execution of the four actions *Restore Infrastructure, Inform stakeholders, Prognoses disruption impact* and *Determine and implement recovery plan* (see Figure 4).

Activity Diagrams Formalization in SysML

The whole process of disruption management can be described as an activity diagram, where sequences of actions transform input to output tokens. The input/output pins of actions are connected to enable the flow of inputs/outputs. In addition, the execution of the actions is enabled by control tokens.
The activity diagram of the Disruption Management Process (Figure 3 and Figure 4) shows the connection of the actions and the activities composing the disruption management process. In a SysML activity diagram, the activity starts at the initial node shown as a solid circle or triggered by an event (denoted as pennant). A token is placed on that node and triggers the execution of one or more actions via the outgoing control tokens. Using SysML it is possible to model parallel actions, joining threads, conditional actions as well as constrained behavior.

Determining and Implementing A Recovery Plan

Determine and implement a recovery plan is one of the most important activities of the process. It is triggered by the *Diagnose disruption* action. The activity (Figure 4) takes as input all the known information. The action performed here is the iterative performance of *Evaluate working schedules and emergency schedules quality*, comparing the current working timetable and the emergency schedules proposed by the activity *Support rescheduling*.
If the quality of one of the available schedules is satisfactory, the action *Choose the best schedules, implement and update working schedules with them* is executed, the results are recorded and the activity is terminated. Otherwise, the *Support rescheduling* action is executed again.

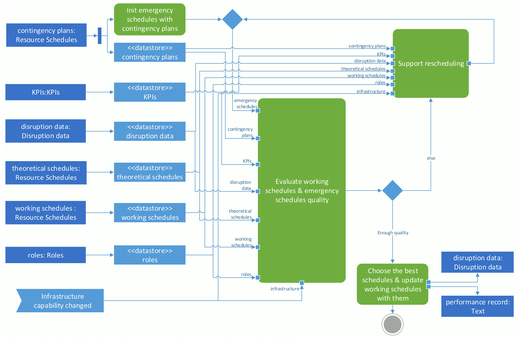

Figure 4: SysML Requirements diagram of rescheduling process.

Model Checking on an Activity Graph

The given activity models can be used to check properties of the system behavior. Therefore the SysML activity diagrams have been translated into a state graph (see Figure 5). The given activities can be shown in a single state graph, but they contain concurrent actions. These result in a few states depicting the termination order of the concurrent actions which are named after the latest terminated action or an action where it is waited for (denoted by a "!" before the action's name). In the resulting graph the upper part (above *evalWS*) represents the activity given in Figure 3. The state graph makes clear that there are some concurrently working threads which sum up in *evalWS*. The lower part of the state graph representing the activities of Figure 4 instead contains some recursions.
Now, the absence of deadlocks and the system's termination can be verified. To ensure the termination of the activity, the quantity of recursive runs has to be restricted. This can be done by number or by time. The number of possible timetables is finite, but in many cases a new feasible schedule will be found without checking all of these, or a timeout will be set to prevent full enumeration.
Except the accepting node *choose best*, each node has at least one following node unequal itself, so the system is deadlock free as long as all processes finish in finite time. Having in mind that these processes often involve human interaction, time outs are needed, as well as standard values for their output objects to proceed (or abort) the system run.

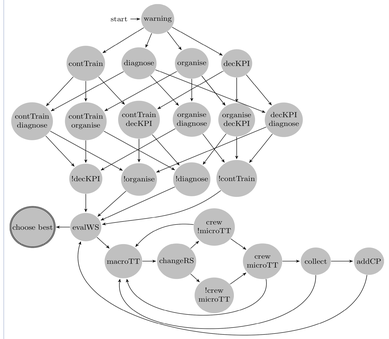

Figure 5: State transition graph of disruption management process.

Summary

In this work, we have formalized the disruption management process implemented by different railway IMs in Europe using SysML diagrams. Its consistency was shown by validation through a model-checking algorithm. The decomposition of the process into basic requirements and activities allows a fine analysis of the weaknesses of the process and of its possible improvements. First, it has emerged in the discussion with IMs that communication is critical. The ability to view a shared plan, or to distribute information electronically rather than verbally, would benefit the whole disruption management process. Furthermore, an automation of the different tasks is envisaged by most IMs. Through the decomposition of the overall activity the formalization presented may help the stakeholders to move in this direction.

Chair of Traffic Control an Process Automatisation
Prof. Dr.-Ing. Jürgen Krimmling
phone: +49 351 463 36784
e-mail: vlp@tu-dresden.de
web: http://tu-dresden.de/vlp

Acknowledgements
The research at the basis of this work was funded by the European Union Seventh Framework Programme project CAPACITY4RAIL, grant agreement no. 605650.