

LUKS – ANALYSIS OF LINES AND JUNCTIONS

*David Janecek, Institute of Transport Science, RWTH Aachen University,
janecek@via.rwth-aachen.de*
*Frédéric Weymann, Institute of Transport Science, RWTH Aachen University,
weymann@via.rwth-aachen.de*

ABSTRACT

In railway operations research several approaches exist to validate and analyse the consequences of infrastructure and/or timetable changes. It is possible to classify the used methods into analytical, simulative or compilatory methods. The appropriateness of an approach depends on the actual situation.

This paper introduces the software tool LUKS which combines all three approaches. Based on former tools developed at the Institute of Transport Science, LUKS incorporates the current state of the art in this scientific domain.

Keywords: analytic, infrastructure, railway operation, simulation, software tool, timetable compilation

INTRODUCTION

In the context of railway operations, infrastructure and timetables are deeply interconnected. Any change in the infrastructure can affect the quality of service, or may even result in an operational unusable timetable. On the other hand, updates of the timetable may result in unacceptable infrastructure utilisation, leading to the necessity for trackage to be redesigned.

As infrastructure changes are time and cost-intensive, various methods exist for evaluating the quality of either an infrastructure layout or a timetable in relation to one another. Various approaches are feasible depending on the accuracy of the data available. In the case of long-term analysis, where the only traffic-flow information at hand is somewhat imprecise, queuing theory methods are used. Mid-term analysis, which considers individual infrastructure changes or compares different possible timetables, is conducted using simulatory approaches. To conclude, short-term issues such as minor changes to the current schedule are tackled using compilatory methods. Dedicated software tools exist for each of these approaches.

This paper introduces the LUKS software tool (an acronym for the German term “Leistungsuntersuchungen von Knoten und Strecken” – analysis of lines and junctions), which is to the best of our knowledge the first integrated tool for all three approaches. Based on detailed track and station layouts and a more or less accurate timetable, the different LUKS modules help infrastructure experts to determine the possible quality of service for a certain time horizon. Thanks to the sophisticated estimation of running and blocking times, timetabling becomes an interactive task with real-time conflict detection. The analytical constituents of LUKS are based on the established ANKE und SLS tools and use various queuing theory methods. The built-in simulator adapts the BABSI simulation tool to the current state of scientific knowledge.

DATA MAINTENANCE AS THE BASIS FOR ALL INVESTIGATIONS

The LUKS software tool was developed in response to the plethora of programs available hitherto and the drawbacks this entails for the inputting and use of timetable and infrastructure data from the operative systems run by DB Netz AG. In the past, for instance, simulation and analytical computation of one and the same junction has involved acquiring infrastructure and timetable data separately in different systems. This led DB Netz AG to join forces with the Institute of Transport Science at RWTH Aachen University with a view to developing a new tool that was additionally to be rendered state of the art.

LUKS is modular in design and comprises several core components, which are adumbrated below, plus modules with which to compile timetables, subject lines and junctions to analytical examination, and simulate the timetabling and operating processes.

Modelling railway infrastructure

Underpinning all studies is a representation in model form of infrastructure planned or available. The SPURPLAN track-layout software originally developed at the Institute [1] is used as the basis for this. LUKS features a number of additional infrastructure elements to map, for instance, all automatic train control systems in use in Germany now or in the future (PZB 90, LZB or ETCS Level 1, Level 1 LS, Level 2 and Level 3). Infrastructure is modelled with metre-level accuracy to deliver a database that is sufficiently detailed to facilitate exact computation of running and track-occupation times.

Infrastructure Editor

It is usually the case that investigations of railway operations reveal a need to alter imported infrastructure data speedily and straightforwardly. But it must also be possible to completely recapture infrastructure without unreasonable effort.

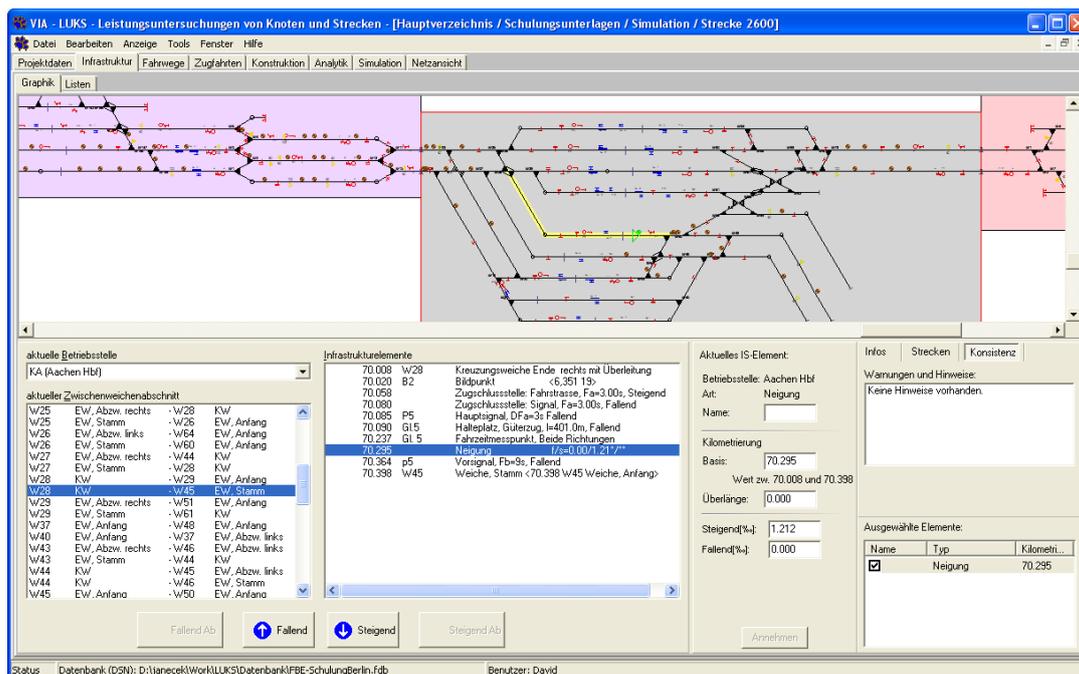


Figure 1 – LUKS Infrastructure Editor

Figure 1 shows the Infrastructure Editor’s standard view with the usual screen layout for LUKS: tracks can be seen on a one-to-one basis in the upper half, whilst details of the program module currently running are to be found in the bottom half.

In the case of the Infrastructure Editor, these include a list of infrastructure elements within the inter-S&C section currently selected, details of the current track-layout element and a number of buttons for rapid adjustment of the element’s values and the graphic display.

Great value was attached when developing the user interface to simplifying standard work steps and providing access to them within a low number of operator actions. The various Editors are accordingly configured in context-related fashion; little resort is had to dialogue boxes, which make processing rather heavy-going.

Routes and tracks

Building on a given infrastructure, use is made in LUKS of “control post routes” to map all running options admitted by the signalling system. A route of this kind defines the course of a train’s movement through a control post: starting at one control post boundary, the course of the route is described as a series of decisions taken at all S&C systems to be negotiated up to the next control post boundary. Alternative starting and terminating points are stopping berths, track ends or buffers.

It is necessary in order to map ATC equipment in this model to enter routes in such a way that account is only taken of running options that signalling will allow in practice.

Platform tracks can also be entered in LUKS along with routes. They are likewise based on the infrastructure initially captured and are required as a means of demarcating “track

groups” in analytical studies or, in simulation exercises, of establishing spatial limits for various types of interconnection between train movements.

Modelling the operating schedule

Operating schedules are generally subject to a far greater degree of change within a given period than the infrastructure data upon which they are based. It is all the more important, therefore, that there be scope for re-utilising as many of the data already available as possible. For this reason, LUKS is capable of importing data from a variety of science of railway operations (SRO) tools.

Given that LUKS can be used both for analytical studies of lines and junctions and to simulate timetable compilation and traffic processes, distinctions are made between various types of “train objects”, which are managed along hierarchical lines in a tree structure (cf. Figure 2, left-hand side). There exist train families, model trains, single trains, and sub-trains.

Sub-trains correspond most closely to what is understood by a train movement. Key properties are a train’s characteristics and its route. It may, however, occur that a train movement cannot be represented by a single sub-train. This is the case, for example, on the Cologne-Mannheim route via Frankfurt am Main Hbf, a station that is left with the train’s direction reversed. Historical reasons (compatibility with existing data) dictate that any one control post may only appear once in a train’s itinerary, which is why a service of this kind is represented by two sub-trains (one from Cologne to Frankfurt am Main and another from Frankfurt am Main to Mannheim).

The two sub-trains nevertheless form part of a single train movement; as a result, LUKS also features what are known as single and model trains, with the aid of which sub-trains representing a single train movement are managed as one object.

The two train types differ in terms of the subsequent use to which they are put: single trains are needed for timetabling and simulation. Each single train represents an actually occurring train movement and therefore has a concrete time of departure.

Model trains are needed for analytical studies. They are a synthesis of several (similar) single movements into one model movement. Hence, a model train represents one or more single trains, something that can be mapped by, for instance, a daily traffic distribution curve or a list of departure times.

The link between single and attendant model trains is established in LUKS by means of what are referred to as train families. A train family may contain any number of single trains that should ideally have features in common (clockface services, fixed-line services). It may also contain one model train at most that, in analytical computations, represents all single trains.

It is possible, therefore, to imagine a train family as being a clockface or fixed-line service involving several concrete train movements (single trains) and represented by a specially designated train (model train).

Whether one or the other type of train is required depends on the nature of the SRO study being conducted, and LUKS accommodates this by being capable, by means of a train generator, of creating all single trains represented using the model train data available. The generator analyses the operating schedule for the model trains entered and uses it to produce matching single trains with concrete times of departure.

This enables model-based analytical assumptions to be verified by simulatory means. It is additionally possible to compile a viable running schedule speedily using a small quantity of data. As a result, capturing a fixed-line service running at half-hourly intervals over a ten-hour period does not entail capturing all 20 train movements individually. Instead, it is merely necessary to enter a single model train with a corresponding operating schedule. The train generator then uses this to automatically create all twenty actual train paths.

There is additionally scope, following generation, for manually altering one or more of the train paths produced. It is possible in this way to, for instance, shorten the route or change the means of traction with little effort.

Train characteristics

A detail of importance for a train movement besides the vehicle's properties (train name, class or type etc.) concern the latter's dynamics and route.

As well as selecting a traction unit from a suitable database, it is also possible to capture further parameters such as weight, max. speeds or the fit-out of trains with various protection systems in LUKS. The running dynamics side of the Train Editor can be seen in Figure 2.

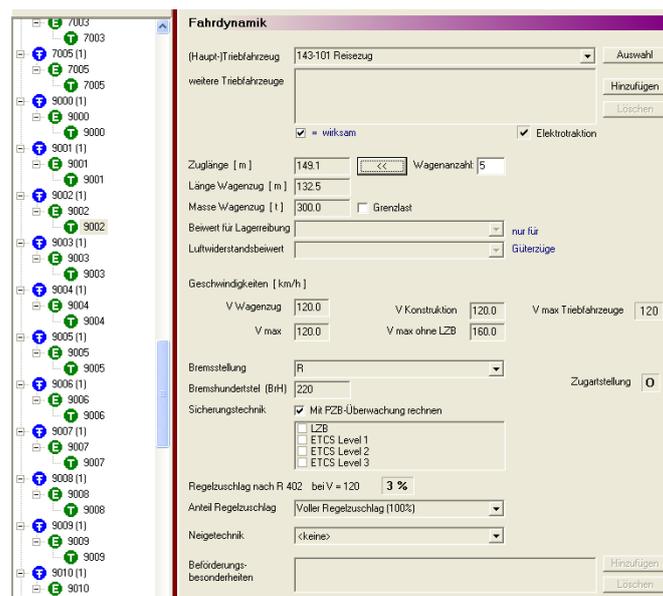


Figure 2 – Running dynamics options in LUKS

Mapping routes

A train movement's route is defined as a succession of routes in LUKS. As well as the sequence of routes completed, it is also possible to capture (minimum) stopping times and both running-time and engineering allowances. The application facilitates convenient capture of such details by LUKS-K, a separate module for the interactive fine-tuning of schedules.

Calculating running and track-occupation times

LUKS is capable on the basis of the infrastructure available and with the aid of the train data captured of accurately calculating running times to the metre and second. The running times calculation utilizes the same algorithms as RuT-K, the standard tool used by the DB Netz AG to compile timetables [2]. In the course of first applications, the match of both calculations was verified. Besides a tabular breakdown of results, LUKS also supplies a graphic visualisation in the form of a speed and gradient band.

Figure 3 shows such a band embracing a train's route together with its max. permissible speed under the signalling system in operation (purple: ATC system like PZB, LZB or ETCS), the speed profile for the line (black), the train's current speed (blue) and a gradient band (jagged trace shadowing the horizontal red line in the middle).

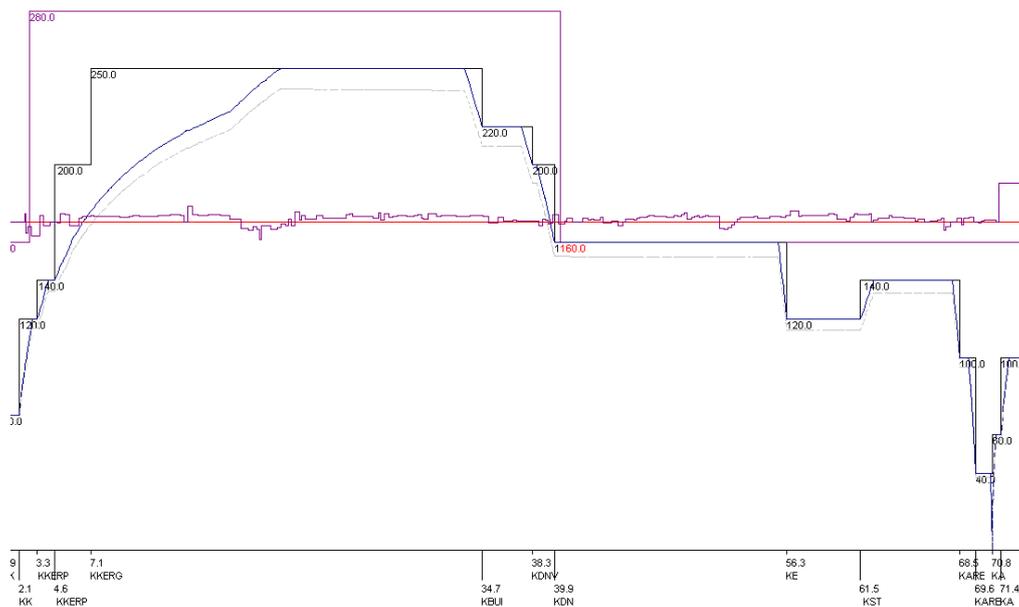


Figure 3 – Speed band for a train movement

Track-occupation is computed on the basis of the running time calculated. This involves forming blocking times along the line negotiated using distance/time curves. Blocking time is taken to mean the time during which a block section between two block signals is occupied by a train, inclusive of various prior and subsequent running times.

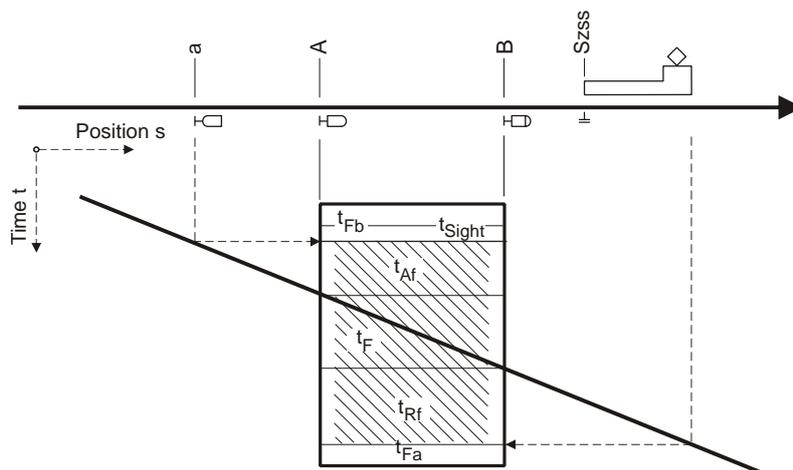


Figure 4 – Element of a blocking time unit

Figure 4 shows a single blocking time unit for a train movement over the block section between signals A and B. It is made up of the following elements:

- t_{Fb} : route setting time (setting points, signals etc.; dependent on interlocking system)
- t_{Sight} : sighting time (the driver must, before passing, be able to clearly identify a “Next signal clear” aspect on the distant signal so as not to initiate a braking application.)
- t_{Af} : approach time (movement between distant signal a and stop signal A)
- t_F : running time (movement between signals A and B)
- t_{Rf} : clearing time (movement between signal B and attendant rear-integrity proving point plus train length)
- t_{Fa} : route release time (by analogy with route setting time)

As with the running time calculation, the blocking time units calculated by LUKS and RuT-K were found to be identical. A blocking-time series is the sum of all blocking-time units along the line negotiated. Further details are to be found in [3].

The distance/time curves and blocking-time series for all trains computed are represented in a track-occupation graph that also points up any conflicting moves. Such conflicts always arise when several train movements wish to use the same infrastructure simultaneously. They can be remedied either directly in the running-path definition, in the course of further processing, or when the timetable is fine-tuned.

SRO METHODS BUILT INTO LUKS

LUKS supports science of railway operations (SRO) studies on the basis of compilatory, analytical and simulative methods and can thus be adopted for long, medium, and short-term planning as illustrated in the sections that follow.

All methods operate with the same base data (environment settings, master data, infrastructure), which makes it easy to compare differing approaches. It additionally cuts working times and avoids potential error sources that might otherwise arise owing to

- data exported from and imported to another program
- duplicate post-processing of such data in various programs
- different settings, and
- differing degrees of computing accuracy.

LUKS, by contrast, immediately makes all settings and adjustments available to the three SRO methods supported.

LONG-TERM PLANNING: LUKS-A

Area of adoption for analytical methods

Analytical methods are suitable for obtaining universally applicable information on usage levels for sections of infrastructure or on the way such sections respond to increases in the use of individual routes [3, 4]. One advantage of analytical modelling is that it rapidly detects infrastructure pinchpoints by identifying heavily used switching zones or track groups.

Speed of examination is particularly crucial in the case of long-term planning, which often involves comparing differing infrastructure variants in combination with several operating schedules. It needs to be borne in mind, moreover, that a concrete timetable will seldom be in place - rather, traffic flows can but be assumed and only rough plans exist regarding the routes to be served.

It is, however, possible to map this uncertain description of future trends in the operating schedule with the aid of diverse parameters such as daily traffic distribution curves and by modelling with random variables. Hence, this method is not dependent on actual timetabling constraints. Instead, individual train movements are merged into “model trains” representing the trains involved. These are always required to share a common route, the same order of service stops and similar train characteristics, amongst other things.

Regular-interval clockface services are a prime case in point: a service of this kind is represented by a single model train and the attendant departure times.

The methods adopted in LUKS are a refinement of the ANKE and STRELE procedures which are widely used for infrastructure planning by the DB Netz AG.

Computing steps

Analytical examination in LUKS begins by dividing the infrastructure into what are known as route nodes. A route node constitutes the largest continuous section of track within which all possible movements conflict. It is possible with the aid of such route nodes to exactly localise pinchpoints in the route node as well as on lines feeding in and out of it.

Drawing on a calculation of track occupation times, minimum headways are computed for all sequence-of-trains scenarios that depend on the blocking-time series and rankings for the trains concerned.

First, potential passing stations are determined on the basis of the trains' parameters and, building upon this, passing sections are likewise determined. Minimum running times are computed for each of these sections and a specified minimum headway is allotted to the route nodes concerned.

For the purpose of determining passing sections, LUKS is capable of automatically adding operational stops and, if need be, of altering trains' routes in such a way that an optimum running time can be posited. In an advance over the existing ANKE method, account is taken of braking and acceleration events when stops are automatically entered.

The subsequent calculation of output capacity uses a queuing model to specify the waiting times arising, a distinction being made between scheduled and unscheduled waiting periods. Scheduled waiting times are such as arise in the course of timetabling when trains are forced to forfeit their requested slots on account of track-occupation conflicts. They can be computed with the aid of a queuing model. Unlike scheduled waiting times, their unscheduled counterparts do not arise during the timetabling process but in the course of traffic operations.

Finally, the values computed are compared with reliable values with reference to prescribed quality criteria. This comparison provides information on the level of use of individual route nodes and their anticipated service quality. It is thus possible to pinpoint heavily and less heavily used sections of infrastructure on the strength of the predefined operating schedule.

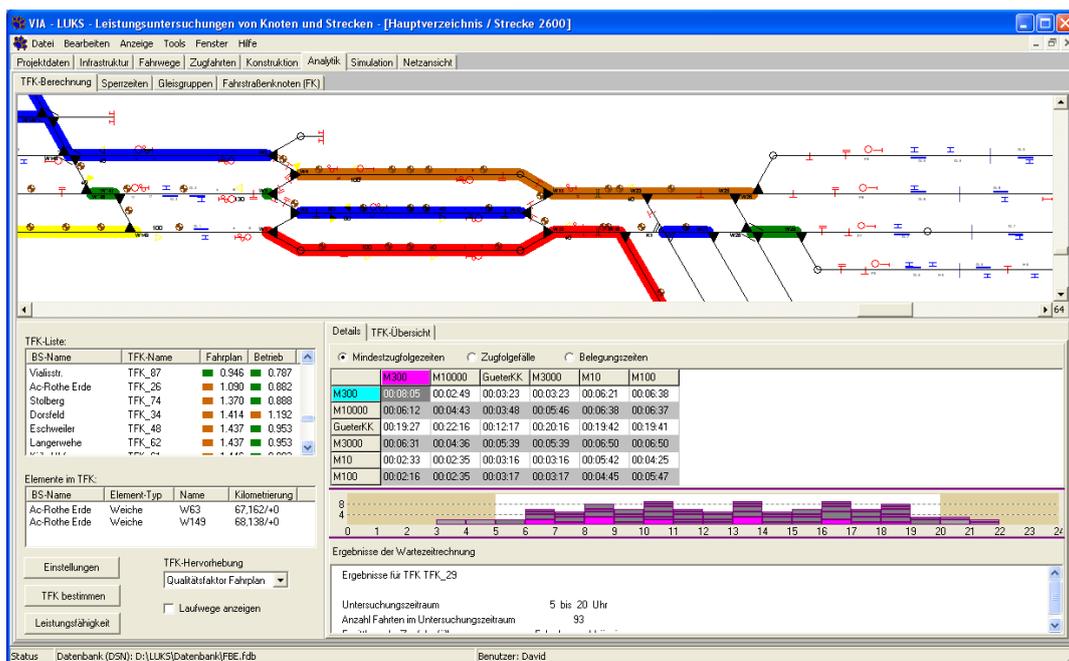


Figure 5 – LUKS during an analytical investigation

Figure 5 shows the user surface for the analytical LUKS module: again, the top half houses the track layout graph on which route nodes highlighted in different colours are to be discerned. The colours make for speedy identification of quiet and busy sections in line with the quality factors computed. The lower half of the screen contains a list of demarcated route nodes together with detailed information on the route node currently selected. Amongst other things, this includes the matrix of minimum headways, a visual representation of the daily traffic distribution curve and detailed results from the waiting time calculation.

MEDIUM-TERM PLANNING BY SIMULATED MEANS: LUKS-S

LUKS-S facilitates simulation of both timetabling and the running of traffic. Both simulation modes operate microscopically on the basis of the infrastructure and train data captured. It is possible with the aid of the two modes to conduct various forms of medium and short-term investigations. The simulation of timetabling provides information on whether a predefined operating schedule is viable, i.e. whether it allows a conflict-free timetable to be compiled that contains the envisaged train movements. Simulation of the running of traffic scrutinises the ability of an already compiled timetable to deal with unforeseen disruptions.

Timetabling interprets existing train movements as being pathing orders and uses the latter to generate a viable timetable with as few deviations from the slots requested as possible. Thus, it realistically maps the path award procedure and constitutes an advance over the BABS method [5]. Paths requested can either be captured in detail by the operator or else automatically derived by LUKS from the daily traffic distribution curve and similar data.

Simulation of the running of traffic involves inputting a conflict-free timetable plus parameters for disruptions. The timetable may be compiled manually by the operator, but there is also frequently scope in medium-term planning for importing a pre-compiled timetable from an external tool. It is also possible to use the outcome of a previous timetabling simulation as a basis. A hybrid, agent-based procedure is adopted under this mode. Conflicts are detected and resolved asynchronously as a means of mapping anticipatory dispatching at operation control centres. Train movements, by contrast, are simulated synchronously so as to be able to map any unforeseen disruptions [6].

The conflict resolution algorithms used in LUKS-S are based on the algorithms of the dispatching tool ASDIS/L, whose concepts were proven in several laboratory and real world tests [7].

Timetable simulation

The simulation of timetabling with LUKS-S proceeds hierarchically. Simulation begins with an empty timetable sheet on which only the highest-ranking train movements are initially entered in their requested slots. All equal-ranking train movements are entered at one time.

Consideration is then given to the first track-occupation conflict to arise chronologically. The overlapping of two blocking-time units is deemed to constitute such a conflict. A number of possible conflict resolution options are identified and assessed, with the following courses of action being contemplated:

- alter the route within one or more operating control points
- alter the sequence of operating control points
- modulate running times
- alter the stopping time
- insert passing and crossing stops
- move entire train path.

Any changes that make sense for either train movement are identified and the additional (scheduled) waiting time they give rise to is called up for evaluation purposes. The option with the best rating is then put to effect. Where it is not possible to identify a valid solution, one of the two train movements is rejected. All conflicts are gone through in this way in chronological order.

As soon as the timetable sheet is free of conflicts, all train movements with the next highest ranking are entered and any fresh conflicts arising are resolved as above. In this way, all train movements are timetabled by rank and a viable timetable is generated on the basis of paths ordered.

Where conflicts arise between train movements of differing rankings, LUKS-S allows for two different modes of procedure. A rigorously priority-based approach will only allow conflicts to be resolved by means of changes to the lower-ranking train movement. Pursuing a policy of partial priority, by contrast, enables changes to be made to the higher-ranking train movement if this is better than the other alternatives. A partial priority approach permits reasonable consideration to be given in model form to the latitude exercised in real timetabling.

Operational simulation

LUKS-S does not adopt the timetable simulation method as a means of simulating the running of services, as the results would be overly optimistic in this case.

Purely asynchronous procedures operate by definition without chronological constraints and allow train movements to be removed at any time. Any disruptions arising are known when simulation begins, therefore, and can be observed in their entirety. Disruptions do not arise unexpectedly as a result, thus being stripped of their most essential quality.

Purely synchronous methods, on the other hand, whilst being capable of portraying the unexpected occurrence of disruptions, are weaker when it comes to mapping anticipatory, wide-area dispatching.

So as to harness the strengths of the two approaches whilst avoiding their various weaknesses in reflecting reality as far as possible, LUKS-S adopts a novel procedure combining the synchronous and asynchronous approaches.

The basic idea is the simulation of two independent agents referred to as “Environment” and “Dispatcher”. The two agents are simulated by two modules working in parallel whose data are kept separate.

“Environment” synchronously simulates the work of the train driver and traffic controller. It monitors the speed of train movements and memorises their current location. “Environment” additionally simulates the setting of routes and signals and incorporates disruptions into the proceedings at random. There is no complex resolution of conflicts at this level; track-occupation conflicts are exclusively resolved by the simulated traffic protection system, i.e. by means of stop/start driving.

The “Dispatcher” agent simulates the work of dispatchers at operation control centres. It does not have access to the Environment data and hence has no precise knowledge of the current status of train movements, being required instead to compute a forecast on the basis of telegrams sent by Environment. Any conflicts arising in the forecast are resolved by asynchronous means. Departing from the current operating situation, therefore, a rescheduled timetable is arrived at that is then forwarded to Environment.

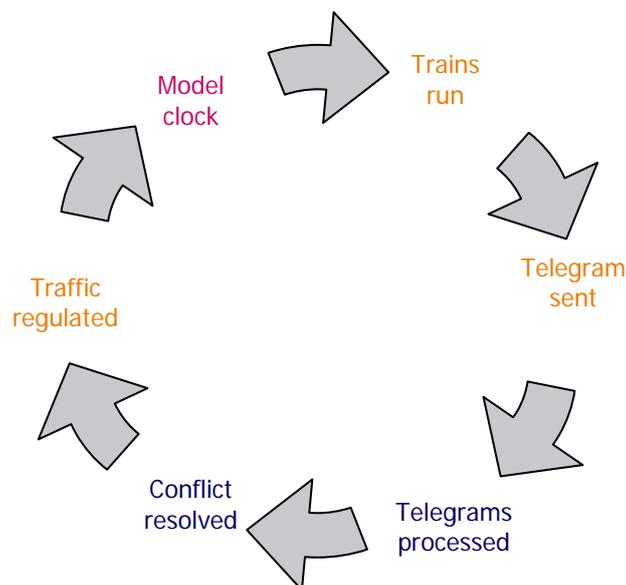


Figure 6 – Course of simulating the running of services in LUKS-S

The running of services consists of a cyclical interaction between these two agents and a model clock as depicted in Figure 6.

The model clocks advances by one second at the start of each cycle. The Environment agent then acts, allowing trains to negotiate the network for one second. It incorporates disruptions whilst at the same time using slack in the schedule to reduce delays.

If any special events such as the restoration of a signal or the commencement of a train movement occur during this step, Environment sends a telegram to Dispatcher.

Dispatcher processes incoming telegrams and arrives at a forecast on their basis. Any conflicts arising are resolved taking account of the time on the model clock. Traffic-regulation decisions such as changes to routes or stopping times are forwarded to Environment as commands. The cycle is completed upon this being done.

It is repeated until the final train movement has reached its destination.

Forming the basis for the running of traffic is a conflict-free timetable. This must be either predefined by the operator or automatically generated with the aid of the timetable simulation facility. LUKS-S also provides the means for conducting dual-tier simulation exercises in which, first, a conflict-free timetable is designed to suit paths ordered and then its robustness is tested by actually running it.

Evaluation

All simulated runs carried out by LUKS-S are archived and can be subsequently evaluated. There is scope for evaluating a single run as well as for pooling a series of such runs.

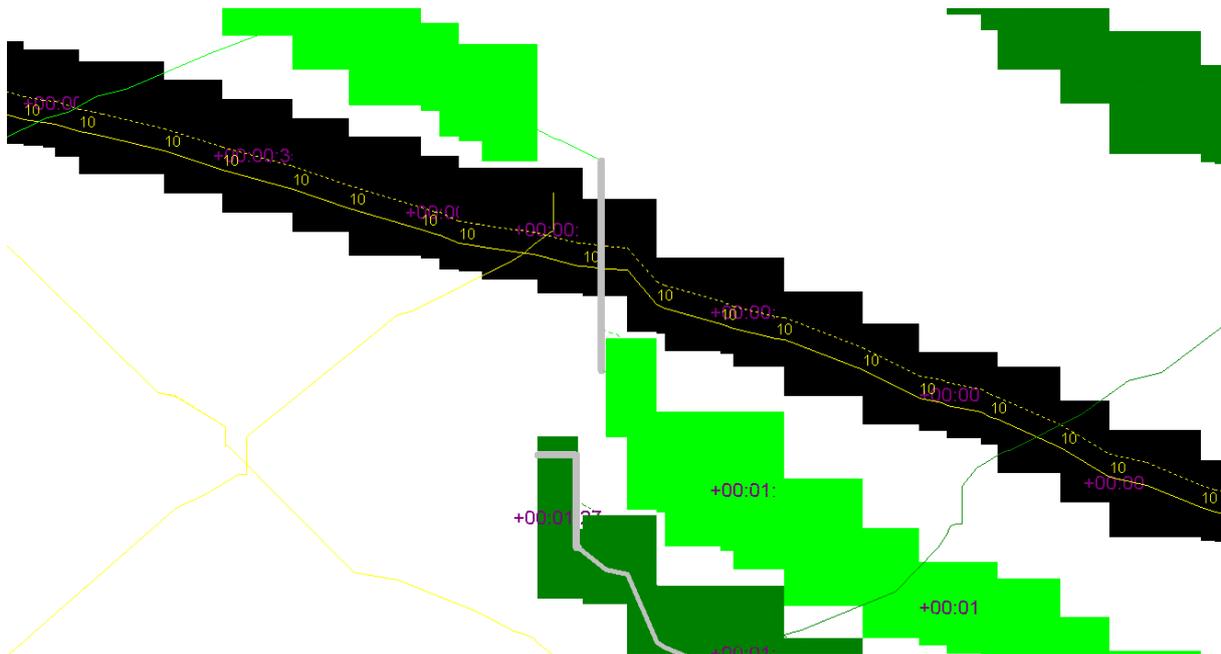


Figure 7 – Comparative timetable sheet

It is possible where single runs are concerned to display the outcome as a comparative timetable sheet (Figure 7). This represents input and output for the run in a single diagram. Output is visualised as a stepped blocking-time series and continuous line, input as a broken line. Changes between input and output are highlighted by means of a bold distance/time curve. The chronological discrepancy is also indicated, furthermore.

Zug	Massnahme	Grund
10	Haltezeitverlängerung in KK um 00:00:35	Konflikt mit 300 bei <KDNV 37,506/+0 HS R201>
300	Haltezeitverlängerung in KDN um 00:01:05	Konflikt mit 10 bei <KDN 41,941/+0 HS A1>
301	Verschieben am Anfang um 00:11:11	Konflikt mit 101 bei <KAS 72,572/+0 W W1>
12	Haltezeitverlängerung in KK um 00:00:35	Konflikt mit 304 bei <KDNV 37,506/+0 HS R201>
304	Haltezeitverlängerung in KDN um 00:01:05	Konflikt mit 12 bei <KDN 41,941/+0 HS A1>
16	Haltezeitverlängerung in KK um 00:00:36	Konflikt mit 306 bei <KDNV 37,506/+0 HS R201>
306	Haltezeitverlängerung in KDN um 00:01:05	Konflikt mit 16 bei <KDN 41,941/+0 HS A1>
102	Haltezeitverlängerung in KK um 00:01:31	Konflikt mit 308 bei <KBUI 30,210/+0 HS 461>
308	Haltezeitverlängerung in KDN um 00:00:33	Konflikt mit 102 bei <KDN 41,941/+0 HS A1>
9100	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9000 bei <KLOV 9,790/+0 HS 205>
9102	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9002 bei <KLOV 9,790/+0 HS 205>
9104	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9004 bei <KLOV 9,790/+0 HS 205>
9106	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9006 bei <KLOV 9,790/+0 HS 205>
9108	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9008 bei <KLOV 9,790/+0 HS 205>
9110	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9010 bei <KLOV 9,790/+0 HS 205>
9112	Haltezeitverlängerung in KK um 00:01:07	Konflikt mit 9012 bei <KLOV 9,790/+0 HS 205>
3000	Laufwegänderung: KARE (A-R-R->A-M-L) KA (A-N3...	Konflikt mit 10 bei <KARE 69,324/+0 HS 9>
3101	Haltezeitverlängerung in KA um 00:06:55	Konflikt mit 101 bei <KARE 68,568/+0 W W4>
3003	Haltezeitverlängerung in KA um 00:02:34	Konflikt mit 3101 bei <KARE 68,568/+0 W W4>
3002	Laufwegänderung: KARE (A-R-R->A-M-L) KA (A-N3...	Konflikt mit 12 bei <KARE 69,324/+0 HS 9>

Figure 8 – List of measures from a simulated run

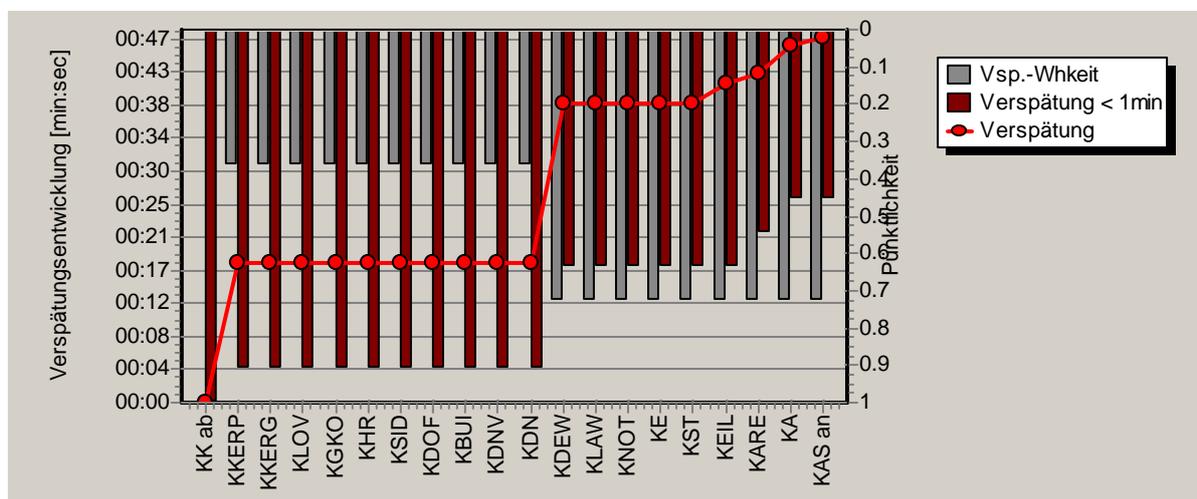


Figure 9 – Visualisation of delay development

Evaluation options further embrace:

- A list of all measures adopted stating the reasons why (Figure 8). This renders the course of the simulation transparent for the operator.
- Visualisation of delay development (Figure 9)
- Tabular enumeration of waiting times and delays upon entry into and exit from the infrastructure
- Tabular enumeration of block section occupancies during the running of services.

SHORT-TERM PLANNING: LUKS-K

The LUKS-K module facilitates the interactive fine-tuning of schedules for train movements entered in the Train Editor and thus provides a convenient means of visualising and processing train movements.

Conversely, the module can also be used within the framework of short-term planning for SRO studies: compilatory methods [8] foresee the operator attempting to resolve any conflicts manually by resorting to the interactive fine-tuning of schedules on the basis of paths requested and, in this way, designing a viable timetable. The timetabling process is thus realistically recreated in this form of investigation, which also allows capacity problems to be detected. The timetable compiled is real, meaning it is necessary to capture all train movements in detail. It is only conditionally possible to make pronouncements regarding usage levels for sections of infrastructure or responses to increases in the use of specific point-to-point routes owing to the coarseness of the data involved.

PROSPECTS FOR FUTURE DEVELOPMENTS

LUKS has been tried and tested in practical applications for the past couple of years and current research and development undertakings are principally focusing on the following spheres:

- Optimising algorithms for memory consumption and hence steadily increasing the potential size of projects
- Extending timetable and infrastructure interfaces
- Localisations for the tool
- Mapping engineering sites and service disruptions
- Computing minimum headways in special cases.

REFERENCES

- [1] Brünger, O. (1995): Konzeption einer Rechnerunterstützung für die Fahrplankonstruktion von Eisenbahnfahrplänen. PhD Thesis, RWTH Aachen.
- [2] Brünger, O. and T. Gröger (2003). Fahrplantrassen managen und Fahrplanerstellung simulieren. In: Proc. 19. Verkehrswissenschaftliche Tage, Dresden, Session 4e (99.1–99.29)
- [3] Hansen, I.A. and J. Pacht (Ed.) (2008). Railway Timetable & Traffic. Eurailpress, Hamburg.
- [4] Ferchland, Ch. and T. Körner (2004). Analytische Verfahren der Eisenbahnbetriebswissenschaft. In: ETR - Eisenbahntechnische Rundschau, 53, Vol. 7/8.
- [5] Gröger, T (2004). Reducing delays by means of computer-aided 'on - the - spot' rescheduling. In: Computer in Railways IX, pages 573–582.

- [6] Weymann, F., A. Kuckelberg and E. Wendler (2008). Coupling of synchronous and asynchronous methods in the simulation of railway operation. In: Proceedings CD of the 10th International Conference on Applications of Advanced Technologies in Transportation (AATT), Athens.
- [7] Kuckelberg, A. and E. Wendler (2008). Real-time asynchronous conflict solving algorithms for computer aided train dispatching assistance systems. In: Computers in Railways XI, pages 555–563.
- [8] Uhlmann, M. and K. Mutschink (2004). Bemessung komplexer Eisenbahninfrastruktur - Die konstruktive Methode. In: ETR - Eisenbahntechnische Rundschau, 53, Vol. 7/8.