

# Distributed system for train route ordering

Karel Greiner and Josef Volek

**Abstract**—The article describes the options of an application for train routes ordering within an annual train timetable by railway undertakings in the territory of the Czech Republic. The application forms a part of a new train timetable information system. A distributed application is in question consisting of a database server, application server and a client program. The client's requirement concerning the data record is executed by the application server in synchronous mode, while individual client data update is carried out in an asynchronous mode by means of an event queue maintained at the application server for each client. The train route can be defined graphically in the rail network window by means of different transport points and sections. Besides the train manipulation function, the application provides different analytical tools and print of the aids for railway staff and passengers. A stage based train formation procedure was designed for the purpose of coordination with the route operator.

**Keywords**—timetable, train, information system, distributed application, application server

## I. INTRODUCTION

The making of train timetables in the Czech Republic has been successfully provided for more than ten years using four information systems. Creation of a train timetable begins in the CEV system [1] in which the railway undertaking enters its requirements for a train track. The railway undertaking subsequently specifies the train engines in the ASO system. Finally, the infrastructure manager calculates in the SENA system [2] the time location of the train and provides additional information. The master data (railway system, engines and wagons) are maintained by EXPERT system [3].

The CEV application works with a local database consisting of binary files. Local databases are synchronized usually at night with the central database via exports and imports. While the users can work with their local train database offline independently of the other users, the changes made by them, on the other hand, will be reflected in the other databases following the data exchange usually the next day. During the

This work was supported in part by the Research Project Theory of Transportation Systems MSM 0021627505.

Karel Greiner is with the Jan Perner Transport Faculty, University of Pardubice, Studentská 95, 532 10 Pardubice, the Czech Republic (e-mail: karel.greiner@upce.cz).

Josef Volek is with the Jan Perner Transport Faculty, University of Pardubice, Studentská 95, 532 10 Pardubice, the Czech Republic (e-mail: josef.volek@upce.cz).

exchange, data may be lost if entered by multiple users on the same day.

Since the commencement of the development of CEV system (1996), computer technologies have moved forward significantly, and therefore in 2006 development began of a new train editor and other applications involved in the creation of the timetable.

## II. NEW INFORMATION SYSTEM

The new concept of the train timetable design is based on two information systems: KANGO and KASO [4]. The structure of these systems and links between them are illustrated in Fig. 1.

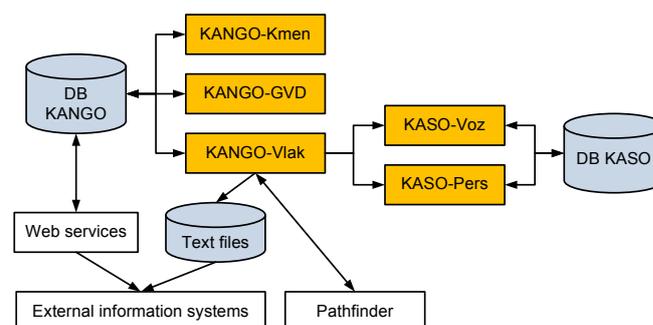


Fig. 1 Information systems for making of train timetables

The KANGO system is primarily designed for the infrastructure manager. It consists of the following subsystems:

- KANGO-Kmen – master data editor – replaces the EXPERT application,
- KANGO-Vlak – train editor – replaces the CEV application,
- KANGO-GVD – serves for construction of the timetable and making of most of print outputs – replaces the SENA system.

The KASO system is intended solely for railway undertakings and contains the following subsystems:

- KASO-Voz – used for production of circulations of engines and rafts of wagons – replaces a part of the ASO system,
- KASO-Pers – used for creating turns of train and locomotive crews – replaces by a part of the ASO system.

KANGO subsystems work over a common central database, which contains a database of master data, trains and users.

Applications KASO-Voz and KASO-Pers read the master data and data from the database of trains through KANGO-Vlak. They use their own central database for information circulation pattern.

The master data and trains data by other information systems operated in the Czech Republic are available in the original and the new format. The original format is represented by text files, the export of which is ensured by the KANGO-Vlak application. The new format is provided through web services. Trains data interchange with the international Pathfinder system [5] is provided by KANGO-Vlak through export and import of XML files.

Below is described the KANGO-Vlak system and its communication with other KANGO and KASO applications.

### III. DISTRIBUTED APPLICATION

Each KANGO and KASO sub-system uses its application server that provides an interface for database access and recorded data control. In the majority of cases, the appropriate application server reads and writes from/to the database only such information data to which it is entitled. The data it uses only for reading, it reads and updates from another application server. For example, KANGO-Vlak application server reads the data on the trains from the database server, while the master data are read from the KANGO-Kmen application server.

The KANGO-Vlak system is developed in the C# programming language. It is executable under operating systems that support the Microsoft .NET Framework platform. It consists of a client program and an application server. Its structure is shown in Fig. 2.

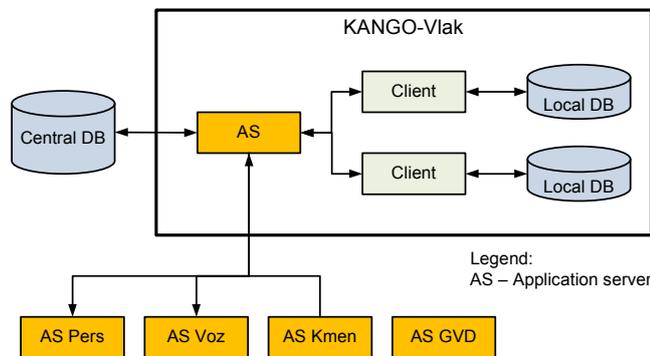


Fig. 2 KANGO-Vlak structure with ties to the other KANGO and KASO parts

The client program keeps a copy of the central database. This copy is stored in the local database. If a client connects to a database for the first time, a new local database is created; otherwise automatic synchronization is performed of the local and remote databases.

The synchronization can be done in two ways:

- Synchronization of the basic tables is made at the forefront upon opening up the database while synchronization of the other tables, especially train routes, is carried out in the background upon opening of the database. When synchronizing the foreground, the user must wait for completion of the operation, while with the background synchronization one can perform almost all operations.
- All the synchronization is performed on the forefront upon

opening the database. This method should be used if a client logs on to the server periodically. If they have not logged in for a longer period, large volumes of changed data may suddenly transfer from the server to the client with no response by the application and thus the user has no information about the synchronization process.

The client application provides an offline mode not connected to the application server. In such case, it reads the data from the local database and does not allow the user to make changes.

The local database is stored in a Microsoft SQL Compact Edition Database Server, which is freely available. The database is encrypted and password protected.

KANGO-Vlak is a distributed application based on Microsoft .NET Remote technology [6], [7]. The client communicates with the applications server in TCP channel via a binary formatter. Such a communication according to [8] is the fastest type compared to HTTP channel communication or SOAP formatter.

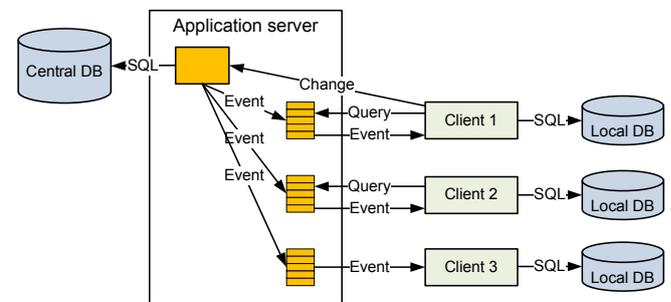


Fig. 3 Adjusted data record and client update

The adjusted changed data record and update at individual clients is demonstrated in Fig. 3. If the client changes an appropriate object, they synchronously send the list of new, deleted and modified entities that have to be recorded in the database to the application server. The application server checks the data integrity. Unless an error is detected, the server transforms the list of entities into SQL commands to change the database. Upon a successful data record, the client may carry on. Then the application server in asynchronous mode files the altered event for each registered client into his server located queue. The change events are taken from the queue by two ways according to the client type:

- If the client is located in a different IP area then the application server (client 1 and 2 in the Figure), the client inquires about the change event and loads it in the stipulated time intervals.
- If the client is located in the same intranet as the application server (client 3 in the Figure), the application server module automatically sends the change events because the client's IP address is accessible from the server. Upon accepting the change event by one of the above means, the client works the change into its operating memory, transforming it into SQL commands and records in its local database.

The clients of the KANGO-Vlak application server include

also KASO-Voz a KASO-Pers application servers.

The train database tables are shared by two systems: KANGO-GVD and KANGO-Vlak. Both these systems directly record the data on the trains to the database and read the data from it. Some tables may be written to only by one of the systems while the other one can only read the data. The other tables may be written to by the both systems, but each only in its records and other records may be only for reading. There are exceptions treated with testing timeliness of the data entered. KANGO-GVD and KANGO-Vlak application servers are not linked. Each gets the information on changes in the train database in their own way.

The KANGO-Vlak application server has a registered notification event handler for Oracle database server, which occurs with a train phase change. The handler reads data on the notified train from the database updates its data in the application server memory and sends the event of change in individual clients.

The KANGO-Vlak application server loads required master data at startup from the application server KANGO-Kmen and registers a handler there in the event of incoming master data updates in of. These events are processed in a similar way as the notification of changes on trains from the Oracle server.

#### IV. DATABASE STRUCTURE

The KANGO-Vlak application includes most of the data available at the timetable planning. The database can be divided into the following groups of data:

- Basic data on the database – such as the date of the period of validity of the timetable, a list of public holidays, a list of changes to the timetable.
- Users – users of the application and their rights.
- User areas of traffic points and sections – any set of traffic points and sections.
- Trains headings – train data independent of the track, such as the number and name of the train, a list of substrates and contents of the freight train.
- Train Routes – a list of traffic points of the train and the following data pertaining to them:
  - general information – such as cancelling a train, special train, percentage of handbrake power,
  - calendar of train,
  - arrival and departure time data, station tracks of the stay and the running tracks,
  - disposition of the railway undertaking to the arrival and departure time data and minimum stay of the train,
  - acts – such as technical check, replacement of the engine.
- Details in the train route related to the section of the train route which may have a defined calendar of its validity within the train trip calendar:
  - transport types of trains – the train specification with respect to traffic regulations,
  - commercial types of trains – a train specification for

passengers,

- railway undertakings,
- ordering party of the train,
- engines,
- locomotive and train crews,
- rafts of wagons – a list of cars, including the order,
- train characteristics – such as weight, length, rolling resistance, braking system, maximum speed,
- notes for individual cars,
- timetable notes for passengers – such as bicycle transport,
- design commands – textual requirements of the railway undertaking to the infrastructure manager.
- Through carriages – are defined as a sequence of carriages or engines in the routes of each train.

#### V. TRAIN EDITOR FACILITIES

The KANGO-Vlak application is primarily designed for editing trains. For this purpose, it includes the following features:

- selection of trains and filtering of the list of trains according to a given combination of parameters,
- creation and cancellation of trains,
- defining and changing a train's route in a graphical view of the railway network,
- editing one or a group of trains,
- copying a train with an option to turn its route,
- connecting trains, the routes of which are successive,
- mass renumbering of trains.

For the analysis of the proposed timetable application, it includes the following tools:

- comparison of two train databases,
- calculation of basic indicators of transport,
- searching the existing and nonexistent connections between trains.

The program allows you to print the following equipment intended for railway staff or passengers:

- Marshalling plan for freight trains,
- Section 3 of Train Formation Plan – List of stations forming shuttle trains,
- Section 4 of Train Formation Plan – List of stations forming balance trains,
- Review of restrictions on the movement of trains,
- Train Guide – information leaflet available for passenger in higher quality trains,
- Overview of higher quality trains – a part of the time schedule book for passengers.

Other application outputs include the following sets for internal use:

- a simplified time schedule,
- list of trains passing through, ending or starting in a selected traffic point,
- list of train stations, where the initial technical check is carried out,

- list of basic data of passenger trains,
- list of occupied and vacant train numbers,
- list of completion of the basic data of trains needed for construction of the timetable.

## VI. USER RIGHTS

KANGO-Vlak system is primarily intended for the railway undertaking for entering requests for train routes. Users are also infrastructure manager's staff, who can enter additional data of trains, browse trains and enter the railway undertakings' trains that do not have KANGO-Vlak program at their disposal.

Each user is assigned with the following rights:

- Type of user – the system provides four types of users: train-maker, railway undertaking, infrastructure manager, and administrator. A "train-maker" type user has basic rights. A "railway undertaking" has the same rights as the user of the "train-maker" type and furthermore can modify the users having a subset of his/her rights. A user of the "infrastructure manager" type has the same rights as the user of the "railway undertaking" type and furthermore can modify the infrastructure manager's data. A user of the "administrator" type has all rights.
- Scope of activities – any set of traffic points and sections. If a train route entry relates to a traffic point, the user may edit it, if this traffic point belongs to its scope. A similar rule applies to the information relating to the train route section.
- List of railway undertakings, whose trains may be viewed and edited by the user.
- List of intervals of train numbers that the user can create and cancel and edit data in the heading of the train.
- List of intervals of train numbers in which the user can edit the data in the route of a train, except for engines, rafts of wagons, locomotives and train crews (hereinafter referred to as train objects).
- List of intervals of train numbers in which the user can edit the entered train objects.

## VII. DEFINITION AND TRAIN ROUTE CHANGE

Route setup and its change can be carried out graphically in the rail network window. The window displays the transport points and sections network and provides for the following functions:

- enlarged and reduced network viewing,
- color highlighting of transport points and sections according to the value or interval of values of a selected database item, e.g. traction-based transport section

highlighting,

- display of the legend to the upgraded transport points and sections,
- transport point search according to the number or name,
- network image adjustment – color, font, headings and transport points and sections size.

Transport point and section that is located in the closest distance to mouse cursor represents current point and section. The operation corresponding to the active mode is carried out by a mouse click for the current point or section. The current point and section are color-highlighted.

In defining a train route, the user can switch between the transport point or transport sections mode. Clicking the mouse e.g. at the given transport point in the starting point mode, this point becomes the starting point of the train and gets color-highlighted.

When setting a new train route, the same method is used to determine the starting and the ending point of the train, while the starting point has to belong to the user sphere of action. Upon setting the appropriate command, the program finds the train route by means of the shortest path and using the Dijkstra's algorithm [9], [10]. Distances between the points are used for the sections evaluation. The calculated route is color-highlighted. In the case of an unsuitable route, a combination of the following transport points and sections can be set up:

- a sequence of points to be included in the route – route points,
- banned points to be excluded from the route,
- banned sections to be excluded from the route.

If there is  $n$  number of route points, the program will look up the route in  $(n + 1)$  steps between individual required route points excluding the banned points and sections.

Designed a route can be further adjusted by adding or removing the route and banned points and sections or by changing the starting or the ending points.

When a part of the existing train route is to be changed, we can define two points of route change in the existing route between which the route is to be changed. In such a case, it is essential to set one or more route points between a pair of the points of route change eventually banned points or sections and then have the route generated. A new route between the two points of route change is found that is highlighted and the rest of the train route stays unchanged.

Users can only change a part within his sphere of action at an existing train. The points and sections outside the user area of action are considered banned in looking up a route in such case.

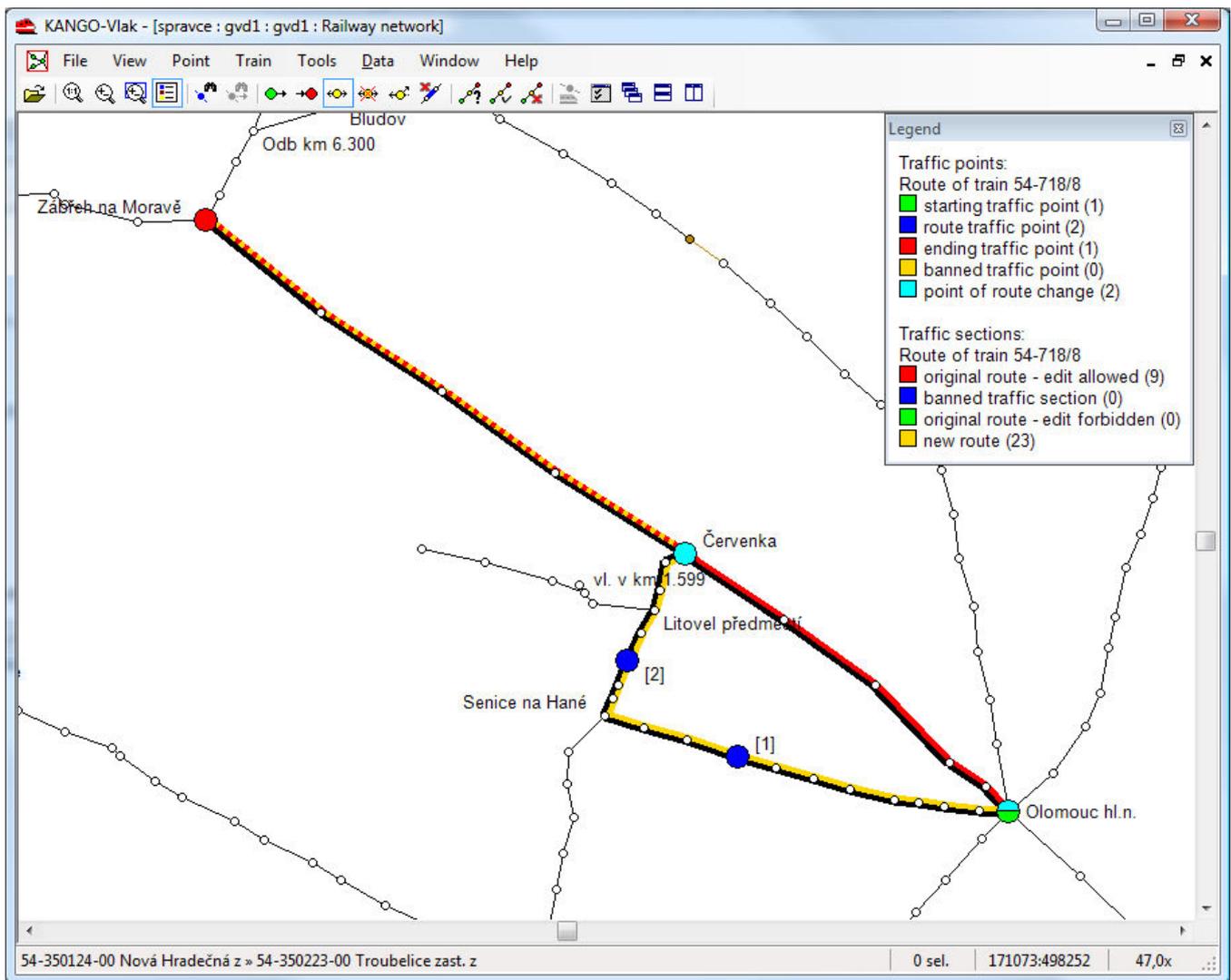


Fig. 4 Graphic scheme of a train route alteration in rail network

Figure 4 demonstrates the alteration of a fictive train route. Color lines show the starting and the ending train point, two points of route change and two route points marked by numbers in brackets. The figure highlights the original and the new train route. The section involved in both the original and the new route is marked with a broken line in two colors. Since the section can be driven in both directions, the lines are parallel. Black color represents the section not driven and colored line represents the train direction. The explanation window contains the list of transport point types and defined train route section including number of their occurrence in the displayed route.

If a train route is to be extended while leaving the original route unchanged, the user can define one point of route change at the original starting or ending point and to determine a new starting or ending point on the train. If a route needs to be cut short, the user can define one point of route change right behind the new starting point or before the new ending point of the train.

## VIII. USER AREAS

User area is any aggregate of transport points and sections. It is used as a user area of action to define the routes or section used by different tool, such as train selection and filtering, looking up connecting trains and transport indicators calculations. This area may contain name and a note.

To prevent the given area from unauthorized adjustments or deletions, it can be locked by the user. In such case it is only the user or the administrator who can make adjustments or deletions. Other users are authorized only to view.

User area transport points exist in the following two types:

- Normal – common points in the area.
- Facultative – a train running over the facultative points in the given area, facultative stopping points and facultative sections:
  - running over the points or sections outside the area is not included,
  - does not run over the points and sections outside the area is included in the area.

- Normal stopping – the train runs over normal stopping points and has to stop at all the points to be included in the area.
- Facultative stopping – the train running over the facultative stopping points has to stop at all the points to be included in the area. Moreover, if the train runs only over the facultative points in the area, facultative stopping points and facultative sections:
  - runs over the points or sections outside the area, it is not included,
  - does not run over further points and further sections outside the area, it is included in the area.

The point of train stopping is the starting and ending point and also the route point where the train stops.

User area transport sections can be divided into the following types:

- Normal – common area sections.
- Facultative – see facultative sections.
- Banned – the train that runs over some of the prohibited sections is not included in the area.

Facultative points, facultative sections and banned sections are used especially in the defining timetable lines, where points and sections exist that form a part of several lines. Figure 5 shows two mock lines: A – E and A – G. Line A – E represents the defined user area. It consists of points A and B as a part of both the lines and the section in between them. To exclude the trains running on line A – G from the A – E, points A, B and the section between them are labeled as facultative. Final points of line A – E must be marked as facultative because they are parts of other lines, at the same time. If there exists a train running only from point A to point B or vice versa, then it would be included in the area of A – E line.

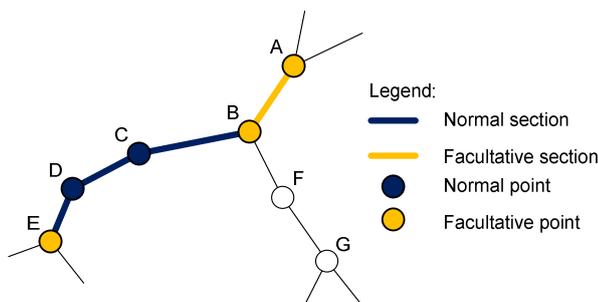


Fig. 5 Mock line A – E user area.

Points of stopping have been introduced for the purpose of defining the timetable lines that form a sub-aggregate of a longer line and do not contain trains running over some points on this line. The area containing the stopping points may not be used as the user action sphere.

Transport points and sections can be added and removed from the area in a dialog window in a tabular format or in the rail network window in a graphic form.

The points and sections within the given area are color-highlighted in the rail network window. Similarly as in the defining a train route, one can switch between the modes of transport point or section type. Clicking the mouse at the given

point/section will:

- add the point or section under an active type unless it is included in the area,
- change its type into active if it is included under a different type,
- remove from the area if it is included as an active type.

Besides this method, a rectangle can be defined by dragging the mouse. A following operations corresponding to an active point/section type and active highlight mode is carried out with the sections and point within the defined rectangle:

- Adding highlighting – points/sections are added into the area as active type, or their type is changed into an active mode.
- Removing highlighting – points/sections are removed from the area.
- Highlighting inversion – an action is taken at each point/section corresponding to the mouse click at such point/section.

If the transport point mode is active, a tool can be selected for the transport points highlighting according to the value or the interval of selected database item value. The same action is taken by means of this tool for the points to be highlighted as if the mouse dragging was selected. E.g. if selected transport points in a given region should be highlighted and the mode “Adding highlighting” is active, then the points of a selected region are added under an active type, or their type is changed into active.

A not-prohibited section can be included in the area only partly. The portion of the inclusion of a section into the area can be defined by a coefficient, e.g. 0.5 of the section’s length. Such function is used in the calculation of transport indicators for a track part whose borderline can be in the middle of a transport section. If a part of the section is included in the area, then only its equivalent is highlighted in the rail network window. To adjust the inclusion of a section part, a special mode is at disposal where the user can switch between the indicated modes of the point/section type. If the mode of section part adjustment is active, then by clicking the mouse at the given section will do the following:

- If such section falls within the area, a dialog window appears enabling the user to adjust the coefficient of the section part inclusion in the user area.
- Else error message appears.

## IX. TRAIN FORMATION PROCEDURE

Train data required by a railway undertaking and the actual data maintained by the infrastructure manager are reported in a pair of trains: the required and the actual. The required train is entered into the KANGO-Vlak system by the railway undertaking as a train requirement to the infrastructure manager. The actual train is made up by copying the desired train and adjusted by the infrastructure manager in the KANGO-GVD system (hereinafter referred to as the designer). The designer can enter the time data in the train route, as well as details of the station and track rails and that is within the

construction field of KANGO-GVD (approximately the area of the Czech Republic). In the traffic points, where the railway undertaking allows it, he/she may also change the sequence of traffic points of an actual train's route.

Data related to a part of the route, which does not belong to the design area of the KANGO-GVD, are loaded into the required train by KANGO-Vlak user directly or imported from the Pathfinder Information System.

The train goes through its formation on to the mentioned stages shown in Fig. 6 and 7. Phases of the required train can be changed by the user who is entitled to create a train.

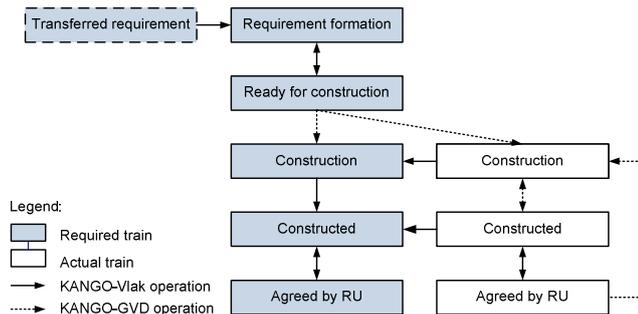


Fig. 6 The basic procedure for forming a train

The procedure for forming a train is as follows:

- The beginning of creating a new timetable is based on data from the previous schedule. The database contains just the required trains of the previous timetable, which are at the *transferred requirement* stage. The actual trains do not exist in the database.
- A user of KANGO-Vlak has an option to create a new train or use a train from the previous timetable. In both cases, such a train gets to the *requirement formation* stage. At this stage, the train from the beginning of its creation or transfer is visible to all users who are entitled to view trains of the given railway undertaking.
- KANGO-Vlak users, who are entitled to edit the train, fill in the details of the train's route, which lies in their scope. At this stage, these users are entitled to enter the train objects as well.
- Users entitled to create a train, may check the box "Access to editing of engines". After checking it, engines can be edited only by a user who is entitled to do so. Unchecking the box may be done by a user who is entitled to edit engines. A user who is entitled to edit the engines, on their entering he/she checks the box "Acceptance of the Engines".
- Similar procedure is used for editing the other train objects. Unlike the engines, there is no checkbox "Acceptance of <train object> with these objects.
- In the event that the required train can be passed on to the infrastructure manager for the construction, an authorized user changes the phase of the train to *ready for construction*. The KANGO-Vlak system allows transition of the train to this phase, if relevant data are filled with and checked the box "Acceptance of the Engines". An authorized user can return the train from the *ready for construction* phase to the *requirement formation* phase and proceed with entering the requirement.
- The KANGO-GVD application server searches required trains at set time intervals in the database, which are at the *ready for construction* stage. It automatically copies the found train to the new actual train that is passed to the *construction* phase and also sets the corresponding required train to the *construction* phase.
- Traffic points of an actual train contain also information about the phase of the train. After creating a new actual train, its traffic points are in the *construction* phase. In the traffic train route points where construction was completed, individual engineers set the *constructed* stage. Once the *constructed* stage is set in all the traffic points of the train route which belong to the KANGO-GVD construction area, KANGO-GVD system automatically sets the stage for the whole train to *constructed*. The designer may later transfer the real train back to the *construction* phase and continue the construction.
- Once an actual train gets to the *constructed* phase, the KANGO-Vlak application server receives the notification from the Oracle database server on a train phase change and imports the data entered by designers in an actual train to the corresponding required train, which it converts to the *constructed* phase.
- If an actual train is in the *construction* phase, the user of KANGO-Vlak can carry on importing the updated data with the chosen required train from the corresponding actual train.
- If the required train is in the *constructed* stage, an authorized user of KANGO-Vlak can change its stage to *agreed by railway undertaking*. At the same time, the KANGO-Vlak system automatically transfers the corresponding actual train also to the *agreed by railway undertaking* stage. At this stage, it is no longer possible to edit the train data. Anytime later, however, the mentioned user may return the pair of trains to the *constructed* stage.
- If the actual train is in the *agreed by railway undertaking* stage, the designers are entitled to change its stage to *construction* and implement the change in its data.
- If the required train is at the *construction* or *constructed* stage, the KANGO-Vlak user can not edit the train. If he/she wants to change its details, an authorized KANGO-Vlak user must first transfer the train to the *requirement change* stage. After completion of the requirement, the authorized user may change the stage of the required train to *ready for construction – change* or return it back the *requirement change* stage.
- Once the KANGO-GVD application server detects the required train in the *ready for construction – change* stage, it automatically copies from the required train all the information, which the railway undertaking is entitled to enter, to the corresponding actual train. After copying, it sets the actual and required train to the *construction* stage. Follows the procedure under point 8.

15. A train can only be deleted by a KANGO-Vlak user that is entitled to introduce the train, with the train being at any stage. In such a case, the required train as well as the corresponding actual train is deleted from the database.
16. Usually, at the end of creation of the time schedule, the user of the infrastructure manager in the KANGO-Vlak application can lock the train. A locked train cannot be edited by any user. The locking can be done at any stage of the train. Unlocking may be done by the user of the infrastructure manager or the administrator.

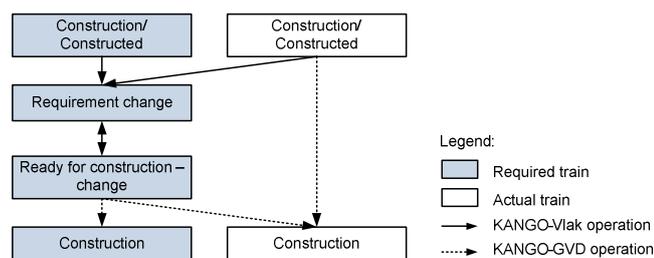


Fig. 7 Procedure when changing the railway undertaking's train requirement

## X. TEST RESULTS

Table 1 shows the values of some properties found in testing the KANGO-Vlak application at a complex train database used in the Czech Republic.

The computers have been configured in the following way:

- Client computer – Intel Core 2 Duo 3 GHz (2 cores), 8 GB RAM.
- Application server – 2x Intel Xeon 3 GHz (4 cores), 16 GB RAM.

No further client was connected with the application server and the computers did not carry any other activities. The application server was located in the intranet of the infrastructure manager; the client was located in the University of Pardubice intranet.

TABLE I KANGO-VLAK SYSTEM TEST RESULTS

Property	Value
Number of trains	17 393
Local and central database synchronization time at the background	15 min
Complex local and central database synchronization time on the front, excluding the changes	20 s
Local database size	150 MB
RAM load with the client database	250 MB
RAM load with the applications server	500 MB

## XI. CONCLUSION

The KANGO and KASO systems are currently in the testing operation. They are expected to be deployed in the routine operation at the end of 2010. Their introduction will significantly improve and accelerate work on the timetable.

Development of the KANGO-Vlak application will go on. It is planned to replace offline data exchange with the Pathfinder system with an online communication via Web services, establish a feature of late request for train route or registration of train routes discounts.

## REFERENCES

- [1] K. Greiner, "Central train editor", Centrální editor vlaků (in Czech), in *Proc. INFOTRANS 2002*, Pardubice: University of Pardubice, 2002. ISBN 80-7194-419-X.
- [2] K. Šotek, et al., *Railway timetable computer aided design. Tvorba jízdního řádu na železnici s využitím výpočetní techniky* (in Czech). Pardubice: University of Pardubice, 2008. ISBN 978-80-7395-137-5.
- [3] V. Tavač, H. Bachratý, and M. Tavač, "Graphic standard data editor – EXPERT", Grafický editor štandardných dát – EXPERT (in Slovak), in *Proc. ŽEL 1997*, Žilina: University of Žilina, 1997, pp. 97–104. ISBN 80-7135-050-8.
- [4] H. Bachratý and K. Šotek, "The concept aimed to railway timetable design innovation", Koncepce směřující k inovaci tvorby jízdního řádu v železniční dopravě (in Czech), in *Proc. INFOTRANS 2009*, Pardubice: University of Pardubice, 2009, pp. 117–126. ISBN 978-80-7395-171-9.
- [5] *Pathfinder – RailNetEurope* [online]. 2010 [cit. 2010-05-24]. Available: <<http://www.railneteurope.com/index.php/pathfinder.50.html>>.
- [6] S. McLean, J. Naftel, K. Williams, *Microsoft .NET Remoting*. Microsoft Press, 2002. ISBN 978-0735617780.
- [7] M. MacDonald, *Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting*. Microsoft Press, 2003. ISBN 978-0735619333.
- [8] D. F. Garcia, C. Uria, J. Molleda, I. Peteira, "Performance comparison of remote method invocation and web services as communication technologies to build distributed applications", in *Proc. 5th WSEAS International conference on multimedia, internet and video technologies (MIV '05)*, Corfu, Greece, 2005, pp. 17–22. ISBN 960-8457-32-7.
- [9] E. W. Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematik*, no. 1, pp. 269–271, 1959.
- [10] A. M. Haziq Lim, N. S. Wan Sazli, B. Hussin, A. A. Azlihanor, S. M. Suhaizan, K. Massila, "A Dijkstra's mobile web application engine for generating integrated light rail transit route", *WSEAS transactions on computers*, vol. 9, pp. 11–20, January 2010. ISSN 1109-2750.
- [11] G. R. Hrin, L. E. Anghel, A. David, "Integrated solutions for freight multimodal terrestrial transport management", in *Proc. 6th WSEAS International conference on applied computer science (ACS '06)*, Tenerife, Canary Islands, Spain, December, 2006, pp. 126–130. ISSN 1790-5117, ISBN 960-8457-57-2.
- [12] P. Cenek, "Object-oriented model of a transportation system", in *Proc. 12th WSEAS International conference on computers*, Heraklion, Greece, July, 2008, pp. 1060–1065. ISSN 1790-5109, ISBN 978-960-6766-85-5.