

METHODS AND TOOLS FOR DESIGNING MODERN TIMETABLE STRUCTURES

Béla Vincze¹, László Kormányos², Viktor Borza³

Summary: An important step on the way towards the railway harmonisation is the introduction of a standardised and attractive timetable structure on the railway networks of the united Europe. Since the design process of such a timetable structure can be very complicated, there is high demand for innovative methods and software tools in this area. This paper describes the basic principles and requirements of a software system, which has been designed to fulfil this demands, and successfully proved its efficiency in a Hungarian timetable harmonisation project.

1. Introduction

In order to provide attractive services for public transportation, including local, regional and international traffic, it is essential to harmonize the timetables for all means of public transportation, so that they can satisfy every demand of the potential passengers. On the other hand, it has no less importance to keep the costs as low as possible. As described in [4] and [5], the most powerful way for creating optimal timetables is to establish variants of the standard European timetable system (Integrierter Taktfahrplan, ITF⁴) on every layer of the transportation network. Depending on the network topology, the available resources and local features, this can be a quite complicated process. However, it is possible to create complex software (CAD) tools which can greatly improve both the designing and benchmarking process of the timetable variants.

In the following sections we will go through the major requirements, demands, experiences, basic methods and some technical details of a software system which have been developed for a Hungarian ITF research project. The aim of the project was to design the backbone of a national ITF system and to design and realise every detail of a suburban ITF pilot project.

¹ Budapest University Technology and Economics, Dept. of Control and Transport Automation
Address: Bertalan L. u. 2., Budapest, Hungary H-1111; Phone: (+36-1) 463-1044
e-mail: egzo@axelero.hu web: <http://web.axelero.hu/egzo>

² Hungarian State Railways (MÁV) Department of Passenger Transport
Address: H-1062, Andrásy út 73-75., Budapest, Hungary
Phone: (+36 1) 322 0660/7459; E-mail: kormanyos.laszlo@szesza.mavrt.hu

³ Hungarian State Railways (MÁV) Department of Traffic Management
Address: H-1062, Andrásy út 73-75., Budapest, Hungary
Phone: (+36 1) 432 3168; E-mail: borzav@mav.hu

⁴ Integrierter Taktfahrplan: Symmetrical regular-interval timetable. Also known as „Intelligent Timetable”, or „Clockface schedule”.

2. Generating the „ultimate” timetable

The ultimate timetable with optimal operating times and connections is the Holy Grail of operational research. Nowadays the fashionable approach to generate the very best timetable leads to the use of state-of-the-art optimizing methods, like genetic algorithms. Initially, we also followed this track, but working in the real life, soon faced astonishing experiences: setting boundary conditions (e.g. train types and their corresponding intervals, priorities etc.) requires a lot of time and fine-tuning; calculations sometimes take ages; vehicle and staff assignment is problematic, any „ultimate” timetable variants requires enormous amount of post-processing work to adapt special local features and last but not least: considering the expenses and the benefits, it’s very hard to determine which variant is better.

On the other hand, having the ultimate timetable is far from being enough. It’s necessary to have statistical indices, cost calculations and various types of timetable charts which can be used for presentations [5].

So, we have chosen a different approach, following the basic rules of ITF [4] and combining the human creativity and inventiveness with a very efficient CAD system called “takt”. This system, rather than calculating the whole timetable, just helps effectively the design process by optimizing smaller elements and solves many other tasks, leaving great freedom for the user.

3. Architecture of the software system

Figure 1 below shows the rough block diagram of the software architecture used in the project. Depending on the startup mode, the **databases** are connected to the main engine via the local file system or a remote server, allowing great flexibility.

The system engine manages all train paths; their corresponding train consists; provides functions for structure design and analyzing and lastly; has numerous import/export procedures for financial calculations, reports and publishing.

Considering the size of the traffic network, the number of trains and consists which has to be handled in the system, **the efficient user interface** has capital importance. The principles of how to create an efficient graphical user interface will be discussed later.

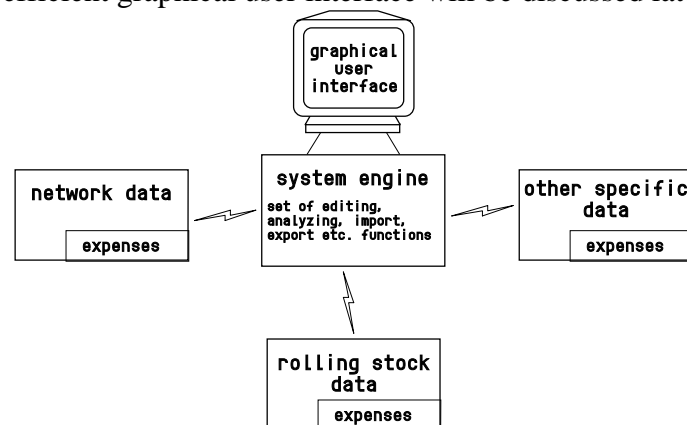


Fig 1. The system architecture

4. Representations of the timetable structure

As presented in [4], the ITF system relies on three factors:

- the periodicity, which has to be a multiple or submultiple of 60 minutes,
- the symmetry,
- the connection system at the network nodes (“spinne” or “spiders”).

It is self-evident that an efficient timetable designing and editing method requires such a representation which is capable of showing all of the three factors concurrently.

Generally a timetable can be represented in two ways: in textual form, using tables or in graphical form with the classical distance-time diagram (Figure 2) [1]. It is obvious that for structure designing purposes only the graphical form can be used.

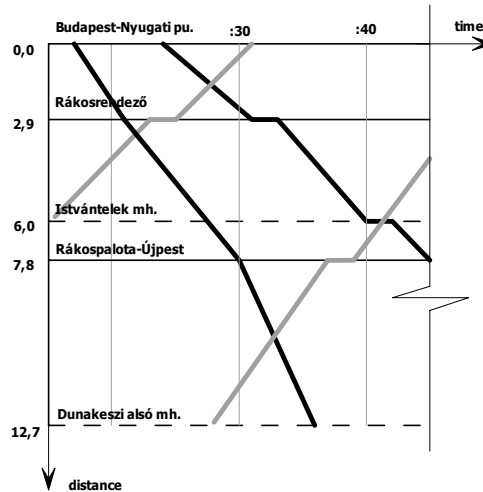


Fig 2. Traditional distance-time diagram representing the timetable

Unfortunately a single traditional distance-time diagram is not suitable to represent an ITF structure, since it can't show the connections at the network nodes properly in all cases. To overcome this major and some other problems, we have developed the following methods for the graphical user interface:

- **Route viewing and editing:** Instead of displaying a single line's distance-time diagram, it should be possible to view the diagram of any route between any points of a network, without limitations on the affected nodes. This also requires an efficient route planner.
- **Working on multiple routes:** It should be possible to switch between route diagrams by a single click, for example, by the means of standard window tab controls.
- **Simultaneous route display:** It should be possible to view two or more routes simultaneously, for example on the upper and the lower part of the display with screen splitting. Any changes made on any of the route diagrams must affect all other route diagrams on the screen. (Figure 3).

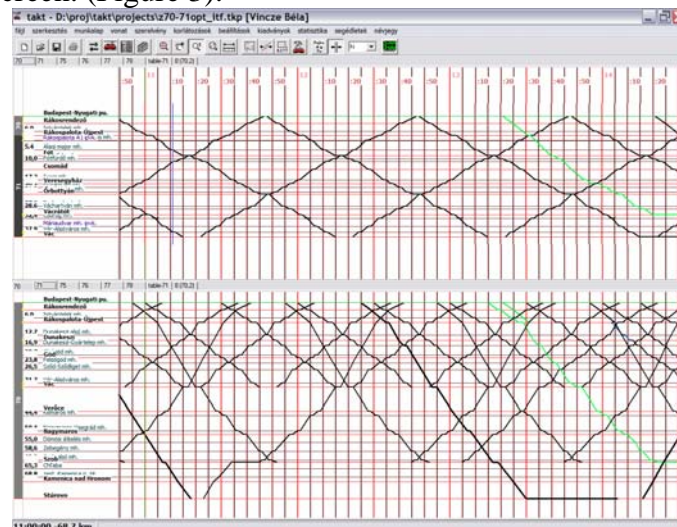


Fig 3. The graphical user interface, displaying two charts of connected routes simultaneously.

- **Displaying the “connection clockface”:** The connection clockface (Figure 4) should show a selected network node’s inbound and outbound traffic on a circular diagram, where the circumference of the circle represents the length of the basic timetable interval (60 or 120 minutes) [2].

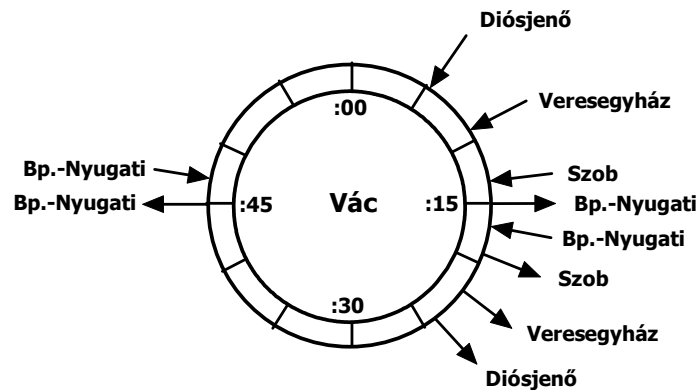


Fig 4. The “connection clockface”, showing the connections at a network node

- **Using 3D graphs:** Two (or more) diagrams should be displayed using three dimensional wireframe graphics, by rendering the diagrams on different planes intersecting at the common nodes. Rotating, moving, zooming this “spatial” diagrams should be done in real-time (Figure 5).

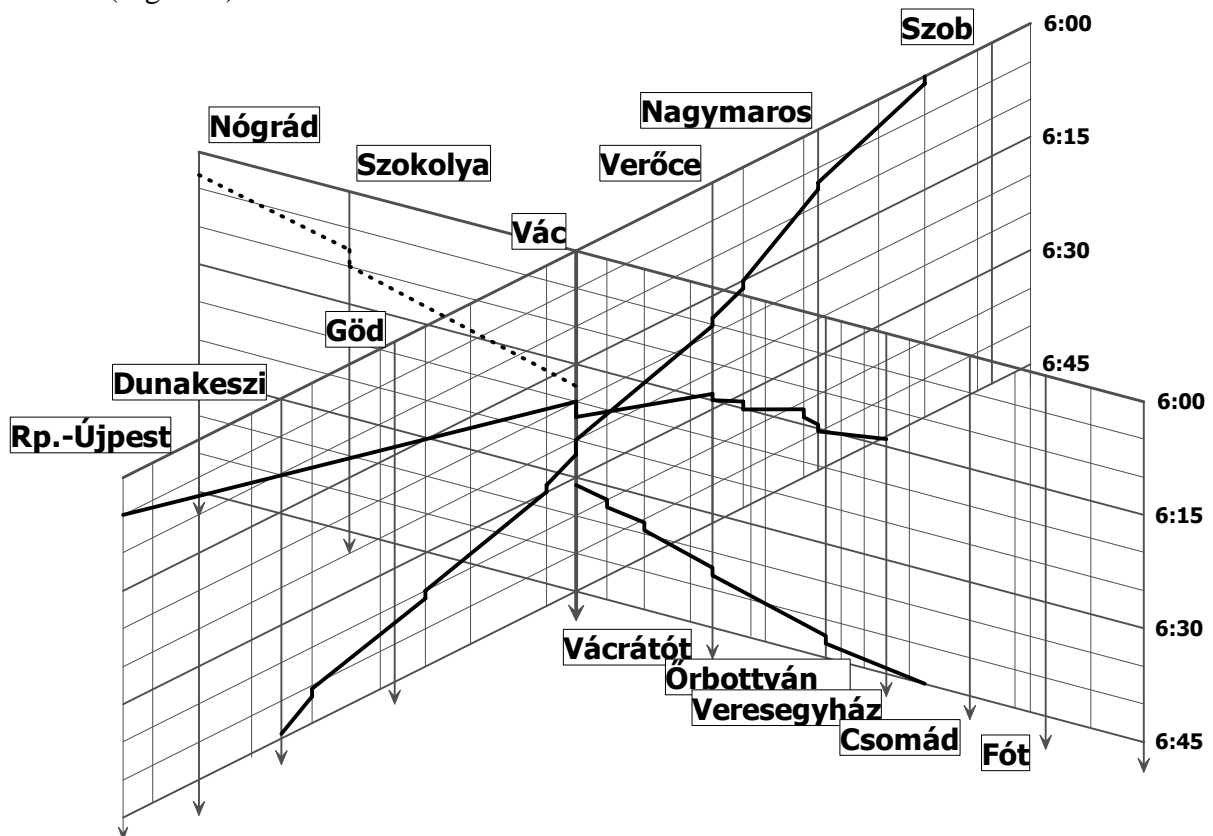


Fig 5. Three-dimensional distance-time diagram showing two lines connected at a common node

- **Displaying the “timetable map”:** The timetable map is a special geographical projection of a periodic timetable. Similarly to the clockface mentioned above, it shows the inbound and outbound traffic of the network nodes. In each direction, arrive/departure times and periodicity information are aligned on the line graphs in textual form (Figure 6).

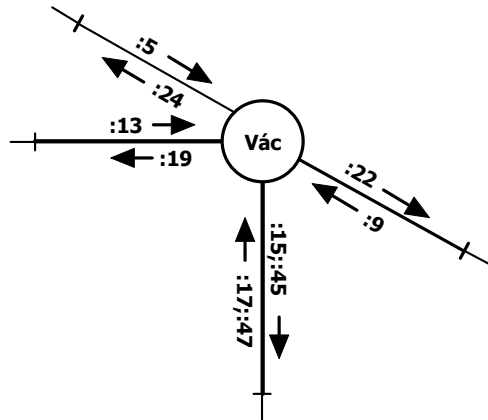


Fig 6. A small section of a timetable map, showing a network node and its connecting lines

- **Drawing station traffic diagrams:** Station track occupancies are needed only in the final stage of the timetable development. For smaller stations, occupancy information can be integrated into the time-distance diagrams, but for larger terminals a separated station traffic diagram should be used [1].

Combining the discussed methods with real-time move, zoom, box zoom etc. functions, we are able to display the timetable in a way which is perfect for ITF representation.

In this representation, vehicle paths are still displayed as segmented lines. Although it could be possible to draw realistic curves based on proper vehicle dynamics calculation, but at the stage of structure design it would be rather confusing and resource demanding than advantageous.

5. Editing the timetable structure

Editing of train paths involves both creating new paths and modifying existing ones. The keys to quick path generation are the efficient route planner and the automatic path data assignment.

5.1. Creating new paths

The route planner should be able to find the optimal route between two points of the network, including any number of intermediate points, based on running time or distance. This requires the network database to contain the distances and running times/speed profiles between nodes. Selection of starting, ending and intermediate points can be done on a graphical map or simple dialogs.

To **convert the generated route to a path**, appropriate data must be assigned to it. Planned stopping points, service numbers, departure times etc. can be set, but for quick work, except for the time data, none of the properties should be obligatory. In order to simplify the user interface, path data setting dialogs must be the same used for data editing. The aim at this point is to fill as much path data as possible automatically with calculated or default values. This requires:

- an implementation of automatic service numbering,
- automatic vehicle assignment,
- automatic running/reserve time calculation (note that it depends on vehicle assignment).

Vehicle assignment can be done with specifying engine types and maximum load, specifying a trainset from the predefined set, or specifying a trainset identifier. To achieve maximal accuracy in benchmarking calculations, the last method (which will be discussed in the following section) should be preferred.

Since ITF is based on periodicity (trains departing in regular intervals) and symmetry, the system should be able to generate not just single paths but path groups and their symmetric pairs. When generating such path groups, it's helpful to classify them with a special attribute that we named „takt-group” for editing operations. A „takt group” contains all paths on a route with regular departure intervals.

5.2. Editing existing paths

Efficient path editing requires sufficient extensions to be applied to the timetable representation discussed above. This can be done in many ways. During our research, we have found that the optimal and „handy” user interface for editing paths is based upon

- definite and easy identification of displayed paths,
- definite and easy identification of places and times along time and distance axes,
- efficient methods for selecting paths or path groups,
- efficient operations on selected paths,
- fast and real-time conflict indication.

Identification of paths is not as easy as it may look at the first glance. Different path types can be shown with different colours or line widths, but it's still not enough. Displaying service numbers next to the path lines can be very confusing, especially when the time/distance scale is small and/or there are many paths within a small area. It is also very resource demanding to draw rotated text. However the problem gets somewhat better when the numbers appear on a separated section of the screen (e.g. on the top, above its path), we have chosen to use „hovering”. Moving the edit cursor over a path causes the path to appear in different colour and display its major properties on a separated section of the screen or in a simple tooltip.

Identification of places and times is much less complicated, as they can be displayed on rulers at the sides of an editing window, aided with a colour grid where grid line colours and widths can represent different attributes. Use of snapping along both axes is essential.

With single clicks or filter operations (by type, by takt-group etc.) paths can be **selected** on the graphical interface easily.

Once the desired paths are selected, we can make various real-time **editing operations** with them. Basic operations are: copy, paste, move, change stopping time, change running time, mirror (for symmetric paths), recalculate running times etc. It's also important to provide functions for changing complex path attributes (e.g. trainset, service number).

The described powerful editing functions can be improved considerably with real-time **conflict indication**. We distinguish two types of conflicts: running time conflict (when the assigned running time is smaller than the lowest possible value), and routing conflict. Both conflict types should be displayed on the user interface by the means of eye-catching colours and/or flashing.

5.3. Vehicle and staff assignment

The presented methods can be also used to sufficiently display and edit vehicle and staff assignments. For example, train paths sharing the same trainset can be connected with

segmented lines on the top and bottom area of the distance-time charts (Fig. 7.). These “connector” lines can be moved with simple drag-and-drop operations.

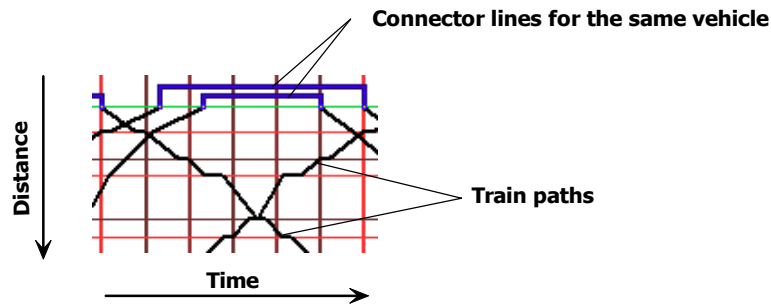


Fig 7. Slice of a distance-time diagram. The segmented lines on the top show that four train paths on the chart use two trainsets.

6. Benchmarking the timetable structure

According to the principles described in [5], benchmarking the designed structure is done in two ways:

- Calculating the **ITF indices**, average service frequency, average running time and display value vector [5] in the reduced three-dimensional vector space for any route. This is very useful for checking whether we reached the optimum value space or not, and to visually compare the quality and value of the timetable variants.

- Calculating various types **statistical and financial data**, like track charges, energy consumption, axle km, seating capacity etc. The accuracy of these calculations heavily depends on the quality of path data assignment. We have found these calculations very important, in order to illustrate that structural changes do not necessarily results in significant increase of costs [3] [5].

Both types of calculations should be done not just for a single day, but for any interval.

7. Optimizing the timetable structure

Although the ITF itself has very strict rules (symmetry, periodicity, connections) [1] [4], there is still room for optimisations on every level of the design process. For example, in the early phase of the structure design it is useful to determine the optimal crossing stations in order to minimize the running time on a single track line.

However, we found that our CAD system with efficient graphical user interface plays the most important role in the fine-tuning stage, when numerous small changes has to made to the structure and the vehicle assignment simultaneously without breaking the rules of ITF.

8. Conclusions

Throughout a timetable development project [5], we have gained many experiences which we have employed in a powerful software tool. The application of the developed software system in the design process made possible to reach our goals and realise a new ITF-compliant suburban timetable structure in every detail.

The most important objective of further development of the tool is to implement a sophisticated module for analyzing the sensivity and recovering capability of the designed timetable.

9. References

- [1] Pahl, J.: Systemtechnik des Schienenverkehrs. 3. Aufl., Verlag B.G. Teubner 2002, Chapter 6.
- [2] Göbertshahn, R.: Der Integralre Taktfahrplan, Vernetzung der Verkehrsträger im Personenfahrkehr als Fundament der Nahverkehrsstrategie der Bahn. Die Deutsche Bahn 5/1993 pp.357-362
- [3] Lichtenegger, M.: Der Integrale Taktfahrplan, Abbildung und Konstruktion mit Hilfe der Graphentheorie, Minimierung der Realisierungskosten. ETR 40 (1991), H. 3-März, pp.171-175
- [4] Borza V., B. Vincze, L. Kormányos: Periodic timetable-map for the Hungarian railway system by the adaption of the European structure, ŽEL 2004, Zilina 2004
- [5] Kormányos L., B. Vincze, V. Borza: Assessment and effective development of timetables adapting value analysis, ŽEL 2004, Zilina 2004