

KAREL GREINER, Ph.D.
 E-mail: karel.greiner@upce.cz
 University of Pardubice, Jan Perner Transport Faculty
 Studentská 95, 532 10 Pardubice, the Czech Republic

Traffic Planning
 Original Scientific Paper
 Accepted: Oct. 17, 2011
 Approved: Mar. 13, 2013

ALGORITHM FOR GENERATING TRAIN CALENDAR TEXTS

ABSTRACT

The article describes a possibility of generating train calendar text for the needs of compiling the annual timetable in the conditions of the Czech Republic. Based on the analysis of the types of texts of calendars that appear in various print outputs, a heuristic algorithm was designed to generate a text from a set of calendar days. The algorithm is a part of an application that also provides a tool to define the text of the calendar by using a mask of sub-periods and calendars to be displayed in them. The algorithm was tested on real data of the timetable. In most cases, the algorithm shows the same or better results than the previously used tools. In several cases, however, a better result can be obtained by the user. The described algorithm to generate the text of the calendar is a part of a program that is used for compiling the timetable for trains in the Czech Republic.

KEY WORDS

railway calendar, railway timetable, generating text

1. INTRODUCTION

One of the basic data of a train is a calendar containing the days it operates. Within the compilation of the annual timetable of trains the calendar includes the days over a period of one year approximately. The passengers and rail workers learn about restrictions on the train running through a text formulation that usually does not contain a simple list of days, but variations of shorter and more meaningful texts.

The role of a modern information system is not only to record a set of train operation days, but also to provide a corresponding textual representation the user would not have to modify for output sets.

The article describes the possibilities of generating train calendar text for the needs of designing the annual timetable in the conditions of the Czech Republic.

2. TYPES OF CALENDAR TEXTS

The annual train diagram in the area of the Czech Republic is compiled using the KANGO information

system [1]. The train calendar is a part of the train data entered in the KANGO-Vlak module [2] by the railway undertaking.

The annual timetable is in most cases valid from a determined Sunday in December of one year to a determined Saturday in December of the next year. It usually concerns the second week in December. The following examples of calendar texts refer to the 2008/2009 timetable period that was valid from December 14, 2008 to December 12, 2009.

The calendar texts appear in various outputs of the KANGO system. The main ones among them are the following:

- Book timetable – the timetable in the form of a book for passengers. It contains passenger trains only. It is created in the KANGO-GVD program and before printing it requires adjustment by the user. The current version in PDF format is available on the Web [3].
- Marshalling plan for freight trains. It is printed from the KANGO-Vlak program without any further adjustment.
- Overview of restrictions on running of trains – an annex to the order to implement the train transport diagram. It includes calendar texts of both passenger and freight transport trains. It is intended for official use only. It is printed from the KANGO-Vlak program usually without any further adjustment.

In the calendar texts there are in addition to individual days of the year also symbols of predefined calendars. They are listed in *Table 1*.

Table 1 – Symbols in calendar texts

Symbol	Meaning
1 to 7	Monday to Sunday
x	Weekdays – Monday to Friday except public holidays
+	Holidays – Sunday and public holidays

Each day of the year in the calendar text is represented by an Arabic number of the day of the month

and a Roman number of the month separated by a period, such as 23.IV.

Public holidays in the 2008/2009 timetable were: 24., 25., 26.XII., 1.I., 13.IV., 1., 8.V., 5., 6.VII., 28.IX., 28.X., 17.XI.

As public holidays are different in each country, for international trains the symbols x and $+$ are not used and the symbols of days in a week are used instead.

Calendar texts can be divided into the following types:

1. The calendar includes all days in the timetable validity period (hereinafter referred to as TVP):
“operates daily”
2. The calendar is empty:
“operates on demand”
3. The calendar includes all days from the beginning of TVP to a certain date within TVP:
“operates until 30.III.”
4. The calendar includes all days from a date within TVP until the end of TVP:
“operates from 1.IV.”
5. The calendar contains most days within TVP:
“no service 24., 31.XII., 1.III. – 13.VI.”
6. The calendar includes a minority of the days within TVP:
“operates 1.V. – 13.VI., 13. – 27.IX.”
7. The calendar includes days represented by a symbol of the predefined calendar or combinations thereof, and alternatively other days (positive days of exceptions) or does not include some days represented by an appropriate symbol of the calendar (negative days of exceptions):
“operates 1 – 5”
“operates x , 6 and 24.XII., 1., 8.V., 28.X., 17.XI., no service 31.XII.”
8. The calendar includes days represented by a symbol of the predefined calendar or combinations thereof, and contains most days of the validity period and also alternatively contains negative days of exceptions – it is used for freight trains only:
“no service 6 and 7”
“no service 6, + and 29.XII. – 2.I., 1.VII. – 28.VIII.”
9. The calendar is composed of sub-periods in which there are days represented by a symbol of the predefined calendar or combinations thereof:
“operates x from 1.VII. until 28.VIII.”
“operates x until 26.III. and from 2.XI., from 30.III. until 29.X. operates 1 – 4”
10. The calendar is composed of sub-periods in which there are days represented by a symbol of the predefined calendar or combinations thereof and one or more periods that include all days:
“operates x until 13.III. and from 2.XI., from 14.III. until 1.XI. operates daily”
11. The calendar is composed of sub-periods of types 9 and 10 and also includes positive or negative days of exceptions:

“operates 6 from 25.IV. until 26.IX. and 1., 8.V., no service 2., 9.V.”

“operates x from 22.XII. until 27.III. and 26.XII., no service 31.XII., from 30.III. operates daily and 14. – 19.XII.”

3. ALGORITHM FOR TEXT GENERATION

A single calendar can have many variations of text representation. For example, the following texts represent the same set of days:

- “operates x until 30.VI. and from 1.IX. and 24., 25.XII.”
- “operates x and 24., 25.XII., no service 1.VII. – 31.VIII.”
- “operates 1 – 5, no service 26.XII., 1.I., 13.IV., 1., 8.V., 1.VII. – 31.VIII., 28.IX., 28.X., 17.XI.”
- “operates 1 – 5 until 30.VI. and from 1.IX., no service 26.XII., 1.I., 13.IV., 1., 8.V., 28.IX., 28.X., 17.XI.”

Which version to select? The algorithm usually selects the shortest version of the text except for some special cases described below.

The main problem related to text generation is the determining of sub-periods. The ideal would be to generate a text for all possible sub-periods and select the shortest possible text. However, the number of variants is very large and it is not possible in real time to explore all the options. Therefore, a heuristic algorithm has been designed.

In literature the algorithm for calendar text generation used on the Slovak Republic railways is briefly described, but it does not solve the cases of consecutive sub-periods [4].

For days (calendar dates) d_1 and d_2 the following relations and operations are defined:

- $d_1 < d_2$ – Day d_1 is older than day d_2 . Similarly, other relational operators are defined.
- $d_2 - d_1$ – The number of days between day d_2 and d_1 . If $d_2 = d_1$, the difference is zero.
- $d_1 + n$ – Day d_1 shifted by n days forward ($n > 0$) or backward ($n < 0$) where n is from the set of integers.

The following notation is introduced for the symbols of the predefined calendars:

b_1, b_2, \dots, b_7 – calendars for symbols 1 to 7;

b^x – calendar for symbol x ;

b^+ – calendar for symbol $+$.

The following text uses the following symbols and functions:

$|A|$ – number of elements of set A ;

D – set of all days within TVP;

C – a set of calendar days, the text of which is to be generated;

B – a set of all predefined calendars

$B = \{b_1, b_2, \dots, b_7, b^x, b^+\}$;

$G(x)$ – a set of days of the predefined calendar $x \in B$ that belong to the set D . For example, $G(b_1)$ includes all Mondays in set D .

$\Gamma(x,y)$ – The set of all days in the period x to y :

$$\Gamma(x,y) = \{d \in D : x \leq d \leq y\} \quad (1)$$

$f(d)$ – Textual representation of day $d \in D$, for example 20.XII.

Sub-periods occurring in the text are represented by a set of sub-calendars. A sub-calendar is an arranged tetrad $[x,y,B^+,B^-]$ where:

x – the starting day of a sub-calendar period;

y – the final day of a sub-calendar period;

B^+ – a set of predefined calendars when the train operates;

B^- – a set of predefined calendars when there is no service.

For the sub-calendar $[x,y,B^+,B^-]$ it is true that

$$x,y \in C, x < y, B^+ \subset B, B^- \subset B$$

The set of sub-calendars is arranged – the elements are sorted in ascending order by x .

The set of days for sub-calendar $s = [x,y,B^+,B^-]$ shall be $H(s)$:

$$H(s) = \left[\bigcup_{u \in B^+} G(u) \setminus \bigcup_{v \in B^-} G(v) \right] \cap \Gamma(x,y) \quad (2)$$

The text representation of a calendar that contains a set of days C and a set of sub-calendars X is provided by function $g(X)$.

The algorithm uses the constants listed in Table 2. The values of the constants were determined by analyzing the texts of the calendars present in the output sets and based on consultations with the users.

3.1 The main part of the algorithm

The procedure to generate a calendar text from set C is as follows:

1. If $C = \emptyset$, the result is the text “operates on demand”.
2. If $C = D$, the result is the text “operates daily”.

Table 2 – The constants used in the algorithm

Symbol	Value	Description
c_1	14	The minimum number of days of the sub-calendar period that may occur in the text of the calendar.
c_2	28	The minimum number of days of the sub-calendar period representing daily operation that will not merge with adjacent sub-calendars.
c_3	13	The maximum difference between two periods of sub-calendars, between which there is no empty week. If the first sub-calendar ends with Monday of the week i and the second sub-calendar begins with Sunday of the week $(i + 1)$, the difference between these days is the value 13.
c_4	28	The minimum number of days of a period between adjacent sub-calendars when there is no train service, i.e. the days of this period does not belong to set C . The constant is used in some stage of merging of the group of sub-calendars.
c_5	70	The minimum percentage of occurrence of predefined calendar days in the sub-calendar, the symbol of which may appear in the text of the calendar.

3. If $\exists b \in B[C = G(b)]$, the result is the text containing a symbol of the predefined calendar. For example, “operates x ”.

4. If the calendar contains all the days from the beginning of TVP until a date within TVP and, at the same time, the number of the calendar days is greater than 2, while the difference of the number of TVP days and the number of the calendar days is greater than 2, i.e.

$$C = \Gamma(\min C, \max C) \wedge |C| > 2 \wedge \min C = \min D \wedge |D| - |C| > 2 \quad (3)$$

the result is the text

$$\text{"operates until" } f(\max C) \quad (4)$$

For example, “operates until 30.III.”.

If the number of the calendar days is less than or equal to 2, the result will be a list of calendar days, for example, “operates 14., 15.XII.”.

If the difference of the number of TVP days and the number of the calendar days is less than or equal to 2, the result will be a list of days that are not included in the calendar, for example, “no service 29., 30.XI.”.

5. If the calendar includes all days from the date inside TVP until the end of TVP while the same conditions are met as in Step 4, i.e.

$$C = \Gamma(\min C, \max C) \wedge |C| > 2 \wedge \max C = \max D \wedge |D| - |C| > 2 \quad (5)$$

the result is the text

$$\text{"operates from" } f(\min C) \quad (6)$$

For example, “operates from 1.IV.”.

6. We create an initial set of sub-calendars S (see Chapter 3.2) taking care that two requirements are met:

that every day from set C belongs to a period of a sub-calendar of set S , i.e.

$$C \setminus \bigcup_{s \in S} H(s) = \emptyset \quad (7)$$

and at the same time, that the sub-calendars in set S do not overlap, i.e.

$$H(u) \cap H(v) = \emptyset \quad (u, v \in S) \quad (8)$$

7. We merge the group of sub-calendars from set S – see Chapter 3.5.
8. The resulting calendar text is obtained through function $g(S)$ – see Chapter 3.7.

3.2 Creating an initial set of sub-calendars

The initial set of sub-calendars S is created as follows:

1. We set $S = \emptyset$.
2. Into the set S we add each sub-calendar $s = [x, y, \{b_1, b_2, \dots, b_7\}, \emptyset]$ that represents the maximal subset of consecutive days of set C and has at least c_2 days, i.e.

$$\Gamma(x, y) \subset C \wedge y - x + 1 \geq c_2 \wedge \nexists [\Gamma(u, v) \subset C \wedge \Gamma(x, y) \subset \Gamma(u, v)] \quad (9)$$

These sub-calendars represent the daily operation period.

3. In set C we go through periods that do not belong in the period of sub-calendars contained in set S . For each such period we go through individual weeks (even partial, with which such a period begins or ends) and we find consecutive weeks in which set C contains the same days of the week (Monday to Sunday). From these consecutive weeks we create sub-calendars $s = [x, y, B^+, \emptyset]$ where x is the first day of the first week of consecutive weeks containing the same days of the week and y is the last day of the last week of consecutive weeks containing the same days of the week.

For the sub-calendar s it is also true that

$$B^+ \subseteq \{b_1, b_2, \dots, b_7\}, \quad H(s) = C \cap \Gamma(x, y) \quad (10)$$

We add these sub-calendars to set S .

After completion of this step it is true that

$$\bigcup_{s \in S} H(s) = C \quad (11)$$

4. For each sub-calendar $s = [x, y, B^+, \emptyset]$ ($s \in S$) for which it is true that $y - x + 1 \geq c_1$, we perform these operations in set B^+ :

- If set B^+ contains calendars b_1 to b_5 and in the sub-calendar period there is the set of days of calendars b_1 to b_5 equal to the set of days of calendar b^x , i.e.

$$\{b_1, b_2, \dots, b_5\} \subseteq B^+ \wedge \bigcup_{i=1}^5 G(b_i) \cap \Gamma(x, y) = G(b^x) \cap \Gamma(x, y) \quad (12)$$

we replace calendars b_1 to b_5 with calendar b^x .

- If set B^+ contains calendar b_7 and in the sub-calendar period there is the set of days of calendar b_7 equal to the set of days of calendar b^+ , i.e.

$$b_7 \subseteq B^+ \wedge G(b_7) \cap \Gamma(x, y) = G(b^+) \cap \Gamma(x, y) \quad (13)$$

we replace calendar b_7 with calendar b^+ .

5. We merge each group of sub-calendars $s_i = [x_i, y_i, B_i^+, \emptyset]$ to $s_{i+k} = [x_{i+k}, y_{i+k}, B_{i+k}^+, \emptyset]$ from set $S = \{s_1, s_2, \dots, s_n\}$ to sub-calendar $s^* = [x_i, y_{i+k}, B^+, \emptyset]$, for which it will be true that

$$y_{i+k} - x_i + 1 \geq c_1 \wedge H(s^*) = \bigcup_{j=i}^{i+k} H(s_j) \wedge B^+ \subset B \quad (14)$$

for $k = 1, 2$

First, we merge adjacent couples of the sub-calendars ($k = 1$) and only then the adjacent triads ($k = 2$). Set B^+ of the merged sub-calendar s^* contains at least one of the pre-defined calendars b^x or b^+ alternatively supplemented with predefined calendars of the days of the week.

There is no point in merging more than 3 adjacent sub-calendars, because there are more than 3 consecutive weeks that include public holidays (for example, Week 1: Sunday 24.XII., Week 2: Monday 25.XII. and Tuesday 26.XII., Week 3: Monday 1.I.).

From the merging we exclude:

- Sub-calendars s_i whose set $B_i^+ = \{b_1, b_2, \dots, b_7\}$.
- Couples of sub-calendars s_i and s_{i+1} that include at least one blank week, i.e. $y_{i+1} - x_i > c_3$.

For example, for the calendar shown in Figure 1, set S will prior to this step contain three sub-calendars:

$$\begin{aligned} & [1.VII., 4.VII., \{b_3, b_4, \dots, b_6\}, \emptyset], \\ & [7.VII., 11.VII., \{b_2, b_3, \dots, b_6\}, \emptyset], \\ & [13.VII., 25.VII., \{b_1, b_2, \dots, b_6\}, \emptyset]. \end{aligned}$$

July 2009							
Mo	Tu	We	Th	Fr	Sa	Su	
			1	2	3	4	5
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31			

Figure 1 – Example of a calendar for creating an initial set of sub-calendars

After this step it will contain only one sub-calendar:

$$[1.VII., 25.VII., \{b^x\}, \emptyset].$$

In the calendar figures, the days of set C are highlighted in bold with a grey background. The holidays are displayed in a frame.

6. For each sub-calendar from set S we perform the algorithm to extend its period – see Chapter 3.3.
7. For each sub-calendar from set S we perform the algorithm to adjust its period – see Chapter 3.4. If the period of sub-calendar $s_i \in S$ has transformed into the beginning of TVP, we exclude from the set S all sub-calendars s_j for $j < i$. Similarly, if the period of sub-calendar s_i has transformed into the end of TVP, we exclude from set S all sub-calendars s_j for $j > i$.

3.3 Extending the period of a sub-calendar

The algorithm is used to extend the period of the sub-calendar from set S , having at least c_1 days. We perform the extending at the expense of the previous or subsequent sub-calendar, whose period is at most one week long. The extension is performed only on days that belong to set C and also belong to a set of the days of predefined calendars of the extended sub-calendar.

The procedure to extend the period of the sub-calendar $s_i = [x_i, y_i, B_i^+, B_i^-]$ ($s_i \in S$) where $S = \{s_1, s_2, \dots, s_n\}$ is as follows:

1. If $y_i - x_i + 1 < c_1$, we do not perform the extension and the algorithm ends.
2. If the S includes the sub-calendar s_{i-1} , we proceed with Step 3, otherwise Step 7.
3. If $y_{i-1} - x_{i-1} > 6 \vee x_i - y_{i-1} > c_3$, we proceed with Step 7.
4. We are looking for the minimum value $d \in C$ in the sub-calendar period s_{i-1} , for which it is true that

$$H([d, x_i - 1, B_i^+, B_i^-]) = \Gamma(d, x_i - 1) \cap C$$

$$(d \in \Gamma(x_{i-1}, y_{i-1}) \cap C) \quad (15)$$

This condition means that all the days with which the period of sub-calendar s_i was extended, and that belong to set C , also belong to the set of the days of the predefined calendars of sub-calendar s_i .

If d does not exist, we proceed with Step 7.

5. The period of the sub-calendar s_i shall begin with day d , i.e. we set $x_i = d$.
6. We are looking for the maximum value $e \in \Gamma(x_{i-1}, d - 1) \cap C$.
If e exists, the period of sub-calendar s_{i-1} shall end with day e , i.e. we set $y_{i-1} = e$.
If e does not exist, we exclude sub-calendar s_{i-1} from set S .
7. If set S does not include sub-calendar s_{i+1} , the algorithm ends.
8. If $y_{i+1} - x_{i+1} > 6 \vee x_{i+1} - y_i > c_3$, the algorithm ends.
9. We are looking for the maximum value $d \in C$ in the period of sub-calendar s_{i+1} , for which it is true that

$$H([y_i + 1, d, B_i^+, B_i^-]) = \Gamma(y_i + 1, d) \cap C$$

$$(d \in \Gamma(x_{i+1}, y_{i+1}) \cap C) \quad (16)$$

This condition has the same meaning as in Step 4.

If d does not exist, the algorithm ends.

10. The period of sub-calendar s_i shall end with day d , i.e. we set $y_i = d$.
11. We are looking for the maximum value $e \in \Gamma(d + 1, y_{i+1}) \cap C$.

- If e exists, the period of sub-calendar s_{i+1} shall end with day e , i.e. we set $y_{i+1} = e$.
12. If e does not exist, we exclude sub-calendar s_{i+1} from set S .

For example, for the calendar shown in Figure 2, before this algorithm is initiated set S will contain three sub-calendars:

- [1.VII., 3.VII., { b_3, b_5 }, \emptyset],
- [6.VII., 24.VII., { b_1, b_2, \dots, b_5 }, \emptyset],
- [27.VII., 28.VII., { b_1, b_2 }, \emptyset]

July 2009						
Mo	Tu	We	Th	Fr	Sa	Su
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Figure 2 – Example of a calendar for the algorithm to extend the period of the sub-calendar

After completion of the algorithm, the period of the second sub-calendar will be extended at the expense of adjacent sub-calendars and the third sub-calendar shall cease to exist:

- [1.VII., 1.VII., { b_3, b_5 }, \emptyset],
- [3.VII., 28.VII., { b_1, b_2, \dots, b_5 }, \emptyset].

The set of predefined calendars of the first sub-calendar additionally contains calendar b_5 that we could omit. However, because it is a short period that will not appear in the final text of the calendar, we can leave it in the set in order to accelerate the algorithm.

3.4 Modifying the period of the sub-calendar

Modifying the sub-calendar $s_i = [x_i, y_i, B_i^+, B_i^-]$ shall be performed in two phases.

In the first phase, we modify the period of the sub-calendar so that the starting and final day of its period belonged to the set of its predefined calendars and also to set C as follows:

1. We set $A = H(s) \cap C$.
2. We set $x = \min A$.
3. We set $y = \max A$.

In the second phase we try to extend the period of the sub-calendar that has at least c_1 days and does not represent the daily operation period as follows:

4. If $y - x + 1 < c_1 \vee (B^+ = \{b_1, b_2, \dots, b_7\} \wedge B^- \neq \emptyset)$, the algorithm ends.
5. If between the beginning of the TVP and the beginning of the period of the sub-calendar there is a period at most one week long that does not include a day that belongs to the set of the predefined calendars of the sub-calendar, i.e. $x > \min D \wedge x - \min D < 8 \wedge \wedge H([\min D, x - 1, B^+, B^-]) = \emptyset$, we set $x = \min D$.

- If between the end of the period of the sub-calendar and the end of TVP there is a period at most one week long that does not include a day that belongs to the set of the predefined calendars of the sub-calendar, i.e.

$$y < \max D \wedge y - \max D < 8 \wedge \wedge H([y + 1, \max D, B^+, B^-]) = \emptyset, \text{ we set } y = \max D.$$

3.5 Merging the group of sub-calendars

The algorithm is used to merge groups of consecutive sub-calendars from set $S = \{s_1, s_2, \dots, s_n\}$, so that the text of the calendar is as short as possible. The procedure is as follows:

- We set $t^* = g(S)$ and $k = 1$.
- We set $m = 1$.
- We set $S^* = S$.
- For each group of sub-calendars $s_i = [x_i, y_i, B_i^+, \emptyset]$ to $s_{i+m} = [x_{i+m}, y_{i+m}, B_{i+m}^+, \emptyset]$ from set S (for $i = 1, \dots, |S| - m$), with the exception of the following groups, we create sub-calendar $\dot{s} = [\tilde{x}, \tilde{y}, \dot{B}^+, \emptyset]$ for the period from x_i until x_{i+m} (see Chapter 3.6). This group must represent a period longer than or equal to c_1 , i.e. it must be true that $y_{i+m} - x_i + 1 \geq c_1$.

The mentioned group must not include sub-calendar $s_j = [x_j, y_j, B_j^+, \emptyset]$ ($s_j \in S$), for which it is true that

$$B_j^+ = \{b_1, b_2, \dots, b_7\} \wedge y_j - x_j + 1 \geq c_2 \quad (j \in \{i, i + 1, \dots, i + m\}) \quad (17)$$

This means that you cannot merge a longer daily operation period with another period.

If $k = 1$, that group must not include couples of adjacent sub-calendars, between which there is a period of longer than or equal to c_4 . This means that we exclude the group if it contains sub-calendars $s_j = [x_j, y_j, B_j^+, \emptyset]$ and $s_{j+1} = [x_{j+1}, y_{j+1}, B_{j+1}^+, \emptyset]$, for which it is true that

$$x_{j+1} - y_j - 1 \geq c_4 \quad (i \leq j < i + m) \quad (18)$$

For sub-calendar \dot{s} we take these steps:

- We set $\dot{S} = (S \setminus \{s_i, s_{i+1}, \dots, s_{i+m}\}) \cup \dot{s}$.
 - We include in set \dot{S} at most two sub-calendars $\tilde{s} = [\tilde{x}, \tilde{y}, \emptyset, \emptyset]$, where $\tilde{x}, \tilde{y} \in C$ and the period of this sub-calendar do not overlap with the period of other sub-calendars in the set \dot{S} .
 - We set $t = g(\dot{S})$.
 - If text t is shorter than text t^* (according to the string length), we set $t^* = t$ and $S^* = \dot{S}$.
- We set $m = m + 1$ and $S = S^*$.
 - If $m \leq 3$, we proceed with Step 3, otherwise we set $k = k + 1$.
 - If $k \leq 2$, we proceed with Step 2, otherwise the algorithm ends.

The algorithm performs merging of at most 4 consecutive sub-calendars. In testing it was found that for

$m > 3$ there is no merging of sub-calendars taking place any more.

3.6 Creating a sub-calendar

The algorithm is used to create sub-calendar $s = [x, y, B^+, \emptyset]$ in the period from u to v ($u, v \in C$), i.e. $u \leq x < y \leq v$. The procedure is as follows:

- We set $B^+ = \emptyset$.
- To set B^+ we add any predefined calendar $b_i \in \{b_1, b_2, \dots, b_7\}$, for which it is true that

$$\frac{|G(b_i) \cap \Gamma(u, v) \cap C|}{|G(b_i) \cap \Gamma(u, v)|} \geq \frac{c_5}{100} \quad (19)$$

- If $B^+ = \emptyset$, the sub-calendar is not created and the algorithm ends.
- If $B^+ = \{b_1, b_2, \dots, b_7\}$ and $\Gamma(u, v) \not\subseteq C$, the sub-calendar is not created and the algorithm ends.
- If $\{b_1, b_2, \dots, b_5\} \subseteq B^+$, in set B^+ we replace calendars b_1 to b_5 with the calendar b^x , if the number of positive and negative days of exceptions when using calendar b^x is less than or equal to the number of positive and negative days of exceptions when using calendars b_1 to b_5 , i.e.

$$|C \setminus P| + |P \setminus C| \leq |C \setminus Q| + |Q \setminus C| \quad (20)$$

where:

$$P = [G(b^x) \cap \Gamma(u, v)] \quad (21)$$

$$Q = \left[\bigcup_{i=1}^5 G(b_i) \cap \Gamma(u, v) \right] \quad (22)$$

- If $b_7 \in B^+$, we replace calendar b_7 in set B^+ with calendar b^+ , if the number of positive and negative days of exceptions when using calendar b^+ is less than or equal to the number of positive and negative days of exceptions when using calendar b_7 , i.e.

$$|C \setminus P| + |P \setminus C| \leq |C \setminus Q| + |Q \setminus C| \quad (23)$$

where:

$$P = [G(b^+) \cap \Gamma(u, v)] \quad (24)$$

$$Q = [G(b_7) \cap \Gamma(u, v)] \quad (25)$$

- We set $s = [u, v, B^+, \emptyset]$.
- We modify the period of sub-calendar s - see Chapter 3.4.

For example, for the calendar shown in Figure 3 a sub-calendar will be created at some point of the algo-

February 2009							March 2009						
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
						1							1
2	3	4	5	6	7	8	2	3	4	5	6	7	8
9	10	11	12	13	14	15	9	10	11	12	13	14	15
16	17	18	19	20	21	22	16	17	18	19	20	21	22
23	24	25	26	27	28		23	24	25	26	27	28	29
							30	31					

Figure 3 - Example of a calendar for the algorithm to create a sub-calendar

rithm for the period from 2.II. until 31.III. In this period there are predefined calendars with the following percentages: b_1 100%, b_2 89%, b_4 and b_5 13%. The percentage limit c_5 is met by calendars b_1 and b_2 that will be included in set B^+ . In this algorithm a sub-calendar [2.II., 31.III., $\{b_1, b_2\}, \emptyset$] will be created.

3.7 Generating text

This algorithm is represented by the activity of function $g(S)$ for the set of sub-calendars $S = \{s_1, s_2, \dots, s_n\}$. First, we modify set S as follows:

- From set S we exclude short sub-calendars $s_i = [x_i, y_i, B_i^+, \emptyset]$, for which it is true that $(B_i^+ = \{b_1, \dots, b_7\} \wedge y_i - x_i + 1 < c_2) \vee (B_i^+ \neq \{b_1, \dots, b_7\} \wedge y_i - x_i + 1 < c_1)$ (26)

- If set S contains only sub-calendars $s_i = [x_i, y_i, \{b_1, b_2, \dots, b_7\}, \emptyset]$, we set $S = \emptyset$.

If $|S| > 1$, or if it contains only one element $s = [x, y, B^+, B^-]$, for which it is true that $x > \min D \vee y < \max D$, the text of the calendar is generated in the following procedure:

- We mark the text of the calendar t and it is empty at the beginning.
- We set $P = \emptyset$.
- From set S we exclude sub-calendar $s_i = [x_i, y_i, B_i^+, \emptyset]$ that has a minimum value x_i and all other sub-calendars $s_j = [x_j, y_j, B_j^+, \emptyset]$, for which it is true that $B_j^+ = B_i^+$. The excluded sub-calendars will be included in set P .
- We determine a set of positive days of exceptions A^+ and a set of negative days of exceptions A^- belonging to the period of sub-calendars included in set P .
- For the sub-calendars from set P we generate the text of the sub-periods, after that we supplement the list of textual presentations of the positive and negative days of exceptions included in sets A^+ and A^- . We add the text to the end of t .
The text of the sub-periods includes at first the beginnings and endings of the periods and then a list of symbols of the predefined calendars included in sets B_k^+ . Set $\{b_1, b_2, \dots, b_7\}$ is represented by the text "daily".
- If $S \neq \emptyset$, we proceed with Step 2.
- We determine a set of positive days of exceptions and a set of negative days of exceptions not belonging to the period of sub-calendars included in set S .
- We generate a text containing a list of textual representations of positive and negative days of exceptions contained in sets A^+ and A^- . We add this text to the end of t .
- Text t contains the resulting text of the calendar.
If S contains only one element $s = [x, y, B^+, B^-]$ ($B^- = \emptyset$), for which it is true that $x = \min D \wedge y = \max D$,

the procedure for generating the text of the calendar is similar to the previously described algorithm with the difference that the period of the sub-calendar is not mentioned in the text. The result is for example, the text "operates x, 6 and 24.XII., 1., 8.V., no service 31.XII."

If the calendar is used for freight trains, we also generate a negative text of the calendar and if it is shorter, it will be the resulting text. We generate the negative text of the calendar for freight trains, if it is true that $b^x \in B^+ \vee |B^+| > 3$ and the set of positive days of exceptions $A^+ = \emptyset$. In such case, we will negate the predefined calendars of set B^+ that we include in set B^- , for example, $B^+ = \{b^x\} \Rightarrow B^- = \{b_6, b^+\}$. Then we set $B^+ = \{b_1, b_2, \dots, b_7\}$ and generate the text, for example, "no service 6, + and 29.XII. - 2.I."

If $S = \emptyset$, the result is the text of type 5 or 6 given in Chapter 2.

4. TEXT CREATED BY THE USER

Since the algorithm generating the text of the calendar from a set of days does not always have to provide a text that is suitable for the users, the KANGO-Vlak program allows the user to define sub-periods and calendars to be displayed therein.

Figure 4 shows a part of the dialog box that is used to enter the sub-calendars. In the "Operates" field we can enter the name of predefined calendars, when the train operates, and in the "No Service" field we can enter the name of predefined calendars, when there is no train service. These fields correspond to sets B^+ and B^- of the sub-calendar. After clicking an appropriate button the corresponding days are marked in the calendar control and the calendar text is generated. For the sub-calendars shown in Figure 4 the result will be the text "operates 2 - 6 from 1.I. until 28.II., from 2. until 31.III. operates x". After generating the text, the user can mark or unmark individual days in the calendar control, adding or removing positive or negative days of exceptions.

	From	Until	Operates	No service
1.	1.1.2009	28.2.2009	2 - 6	
2.	1.3.2009	31.3.2009	x	

Figure 4 - Fragment of a dialog box designed for entering sub-calendars

The actual text of the calendar is read-only. The user can only affect the definition of sub-calendars and individual days included in the calendar.

5. VERIFYING THE ALGORITHM

The text-generating algorithm was tested on real data of the 2008/2009 timetable. Before the introduction of the algorithm described, a tool was used

for creating the text by the user (see Chapter 4) and a simplified algorithm that failed to identify sub-calendars immediately following each other. The simplified algorithm was part of the CEV program [5], which was used prior to the implementation of the KANGO system.

The proposed algorithm was applied to all the calendars contained in the test database. Texts were generated on the computer Intel Core i7 3 GHz with the results listed in Table 3.

Table 3 – Results of generating the texts of the calendars

Indicator	Value
The total number of calendars in the database	2,835
The number of calendars, the text of which was abbreviated	359
The average number of characters, by which the texts of calendars was shortened	20
The number of calendars, the text of which was extended	76
The number of calendars, the text of which was extended and the original text did not contain a daily operation sub-period while the new text does	27
The average number of characters, by which the texts of the calendars was extended to the exclusion of calendars, whose original text did not contain a daily operation sub-period while the new text does	6
The number of calendars with the same length of text, but with a different text	22
The average time of generating a calendar text	5 ms
The maximum time of generating a calendar text	231 ms

The time of generating a calendar text (both the average and the maximum) allows using an algorithm in real time. When marking days in the calendar, the user is not impeded by generating the text, and therefore the text can be generated automatically after each change of the calendar days.

The introduction of the algorithm significantly shortened the texts of calendars that contained multiple sub-periods (on the average by 20 characters).

However, some user-defined calendar texts were shorter (about 3%). Firstly, these are calendars differing in the daily operation period. In the test database there are calendars that do not contain a daily operation period, but the algorithm included such a period in the text. For example:

- User-defined text:
“operates 5 – 7 and 24.XII. – 1.I., 1.VII. – 31.VIII.”
- Text generated by the algorithm:
“operates 5 – 7 until 28.VI. and from 4.IX. and 24.XII. – 1.I., from 1.VII. until 31.VIII. operates daily”.

A part of generating the daily operation period cannot be omitted from the algorithm, because there is a large number of calendars in which there is a daily operation period.

If we exclude the calendars that differ in the presence of a daily operation period, there are 49 calendars whose text is longer when using the algorithm, 6 characters on the average. Thereof, 26 calendars differ in the use of predefined calendars of working days and holidays. The algorithm will use these calendars, if the number of days of positive and negative exceptions is smaller. However, this may result in some cases in a longer text. For example:

- User-defined text:
“operates x, 6 and 24. – 26.XII., 1.I., 12.IV., 1., 8.V., 5.VII., 27.IX., 28.X., 15.XI.”
- Text generated by the algorithm:
“operates 1 – 6 and 12.IV., 5.VII., 27.IX., 15.XI., no service 13.IV., 6.VII., 28.IX., 17.XI.”

The remaining calendars with a longer text (23 calendars) usually differ in confines of the sub-periods. Sub-periods that do not represent a daily operation are primarily determined by full weeks, but sometimes it is better to move the confines. For example, the algorithm provides the text:

- “operates 1 and 2 from 1. until 23.VI., from 29.VI. until 31.VII. operates 1 – 5”,
but more suitable is the text:
“operates 1 and 2 from 1. until 30.VI., from 1. until 31.VII. operates 1 – 5”.

Another problem is the fact that the algorithm provides a daily operation sub-period as the maximum period of consecutive days. If it is immediately followed by another sub-period, it can sometimes be more suitable to include the last day or days of the daily operation sub-period in the follow-up period. For example, the algorithm provides the text:

- “operates from 1.VI. until 1.VII. daily, from 8.VII. until 26.VIII. operates 3”,
but more suitable is the text with sub-periods that begin at the beginning of the month:
“operates from 1.VI. until 30.VI. daily, from 1.VII. until 26.VIII. operates 3”.

6. CONCLUSION

The information system of the train timetable must not only work with the set of train operation days, but it must also provide a corresponding textual representation. The calendar text does not usually contain a simple list of days, but rather variations of shorter and more meaningful texts, which require the use of a certain algorithm.

Types of calendar texts that appear in various print outputs in the Czech Republic were analyzed first. The calendar texts were divided into 11 types.

Based on the analysis a heuristic algorithm was designed to generate a text of calendar. The main task of the algorithm is to determine sub-calendars from which the texts of sub-periods of train running are generated. The algorithm shows in most cases the same or better results than the previously used tools to define the text by the user or the simplified generation algorithm. In several cases, however, a better result can be obtained by the user interference.

The described algorithm is a part of the KANGO-Vlak program that is used for compiling the annual timetable of trains in the Czech Republic. The calendar texts generated by this algorithm appear in various outputs of the KANGO system which include especially the book timetable, marshalling plan for freight trains and an overview of restrictions on the running of trains.

KAREL GREINER, Ph.D.

E-mail: karel.greiner@upce.cz

Univerzita Pardubice, Dopravní fakulta Jana Pernera
Studentská 95, 532 10 Pardubice, Česká republika

ABSTRAKT

ALGORITMUS GENEROVÁNÍ TEXTŮ KALENDÁŘŮ VLAKŮ

Článek popisuje možnost tvorby textu kalendáře vlaku pro potřeby sestavy ročního jízdního řádu v podmínkách České republiky. Na základě provedené analýzy typů textů kalendářů, které se vyskytují v různých tiskových výstupech, byl navržen heuristický algoritmus generování textu z množiny dnů kalendáře. Algoritmus je součástí aplikace, která navíc poskytuje nástroj definování textu kalendáře

uživatelé pomocí masky dílčích období a kalendářů, které se v nich mají zobrazit. Algoritmus byl ověřen na reálných datech jízdního řádu. Ve většině případů vykazuje algoritmus stejné nebo lepší výsledky než doposud používané nástroje. V několika případech však lze zásahem uživatele získat lepší výsledek. Popsaný algoritmus generování textu kalendáře je součástí programu, který je používán pro sestavu jízdního řádu vlaků v České republice.

KLÍČOVÁ SLOVA

železniční kalendář, železniční jízdní řád, generování textu

REFERENCES

- [1] **Bachratý, H., Šotek, K.:** *Koncepce směřující k inovaci tvorby jízdního řádu v železniční dopravě.* In INFOTRANS 2009. Pardubice: University of Pardubice, 2009. p. 117–126. ISBN 978-80-7395-171-9.
- [2] **Greiner, K.:** *Information System of Railway Undertakings Train Track Requirements.* PROMET – Traffic & Transportation. Zagreb: Sveučilište u Zagrebu, 2011, vol. 23, no. 2, p. 141 – 146. ISSN 0353-5320.
- [3] *Knižní jízdní řády* [online]. 2013 [cit. 2013-01-09]. Available from: <<http://www.szdc.cz/drahy/knizni-jizdni-rady.html>>.
- [4] **Bachratý, H., Kršák, E., Sadloň, L., Tavač, M.:** Příspěvek ku generování textových popisů bitových kalendářů. Perner's Contacts [online]. Pardubice: University of Pardubice, 2008, Vol. 3., No. 5. p. 6–15 [cit. 2013-01-09]. Available from: <<http://pernerscontacts.upce.cz/>>. ISSN 1801-674X.
- [5] **Greiner, K.:** *Centrální editor vlaků Českých drah.* In INFOTRANS 2002. Pardubice: University of Pardubice, 2002. ISBN 80-7194-419-X.

