TrainVis 2

Design Document

Bernhard Mecl

October 3, 2012

1 Introduction

TrainVis [MPP07] facilitates the visualization of railway timetables using several different types of representations. Since it is directly based on the low-level toolkits *prefuse* [Hee04] and *TimeVis* [Wei07] much functionality including the data input format is custom to the application and its practical usefulness is mostly limited by missing query and filter operations. *VisuExplore* [RMA+10], on the other hand, already provides an abstraction over those toolkits but is only able to process data representable in a simple tabular form. The goal of the TrainVis 2 project therefore is threefold:

- Generalize VisuExplore to support arbitrary datasets.
- Re-implement the functionality available in TrainVis.
- Enable interaction with the data by way of query and filter operations.

2 Generalizing VisuExplore

2.1 Nomenclature

Since VisuExplore was developed in the medical domain the classes encapsulating the single data table read from a file or a database inherit from the abstract class *Patient*. This class was replaced by the interface *Dataset* in a mostly cosmetic refactoring operation.

2.2 Customizing the creation of visualizations

Even though VisuExplore internally worked with *Patient* objects for historical reasons, many time-oriented datasets can be and actually are already represented using the application's simple tabular data structure where a certain numeric or logic

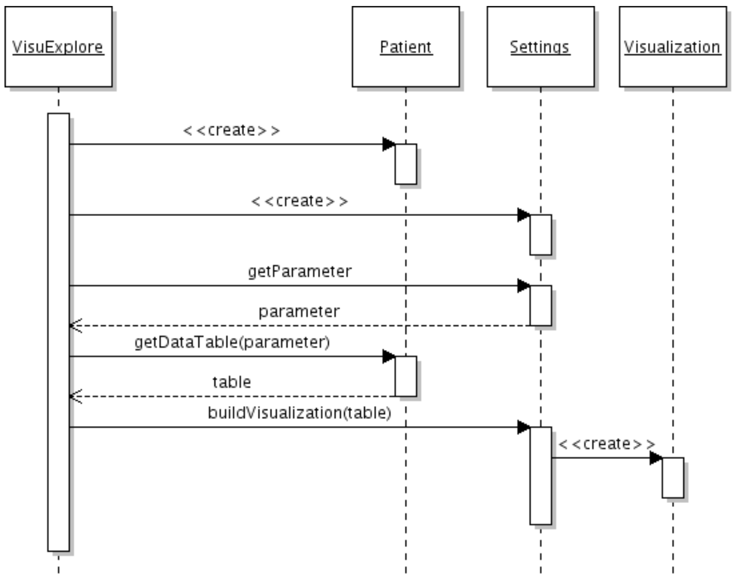


Figure 1: Creation of a visualization object (original)

value is assigned to a parameter at a given timestamp. The main obstacle to the ability of processing arbitrary datasets lies in the tight coupling between this data structure and the creation of visualization objects. In the original design a visualization is created by the main singleton *VisuExplore* object which passes it the one and only data table a *Patient* provides (Figure 1).

The new design shifts the responsibility for creating visualization objects from the main to the settings class (Figure 2). This not only enables the settings object to customize the creation of a visualization but also to pass it several arbitrary data structures provided by the dataset at once. In order to remain backward compatible with existing functionality, the default implementation of the new *buildVisualization* method invokes the original one in the same way as the *VisuExplore* object did before.

2.3 Customizing the creation of standard dialogs

VisuExplore provides a so-called data panel which displays the single data structure in its original tabular form. To adapt this feature to arbitrary datasets, the creation of the *DataPanel* object was relocated to the *TimeVisualiation* class. This way a visualization object may decide on its own how to best represent the available data in textual, tabular, or any alternative form for that matter.

The *NewFacetDialog* class was retrofit to implement the *INewFacetDialog* interface so that datasets may provide their own implementation of this dialog to present visualizations An object implementing this interface is returned by the *Vi-*

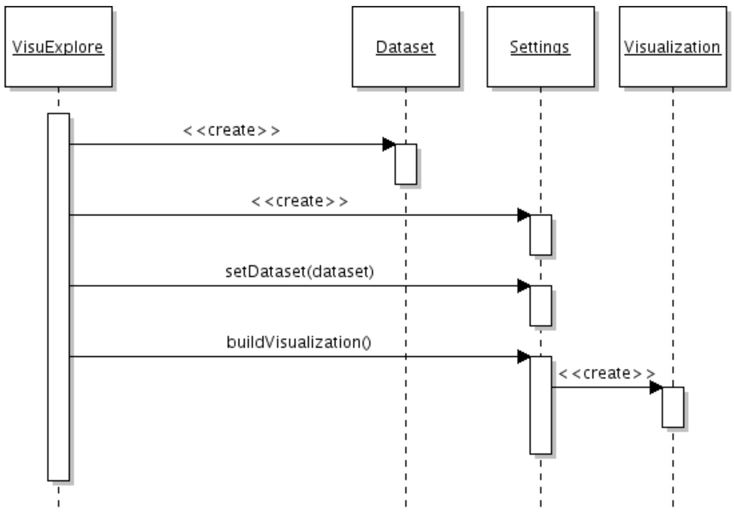


Figure 2: Creation of a visualization object (revised)

sualDataset class which handles all visual aspects of a dataset.

A new toolbar icon launches a dialog which enables the user to interact with the data of the currently selected visualization by way of query and filter operations.

2.4 Multiple document interface

In order to be able to view and compare multiple sets of visualizations at once, the single-window design of VisuExplore using a single timescale has been replaced with an MDI (multiple document interface) where every internal frame has its own independently configurable timescale. Frames may be added, renamed, and removed using the menu bar and their configuration including their respective location and size gets persisted when saving the session.

2.5 Miscellaneous

As VisuExplore's visual padding of at least a month's time is not appropriate for datasets of arbitrary granularity, the dataset itself may now provide a suitable value.

3 Visualizing Railway Timetables

TrainVis 2 re-implements and partly improves the four kinds of visualizations present in TrainVis (see Figure 3). Included are a timetable bar chart with the possibility of switching between absolute and relative timelines and a Marey chart which can be interacted with using the visualization's query dialog.

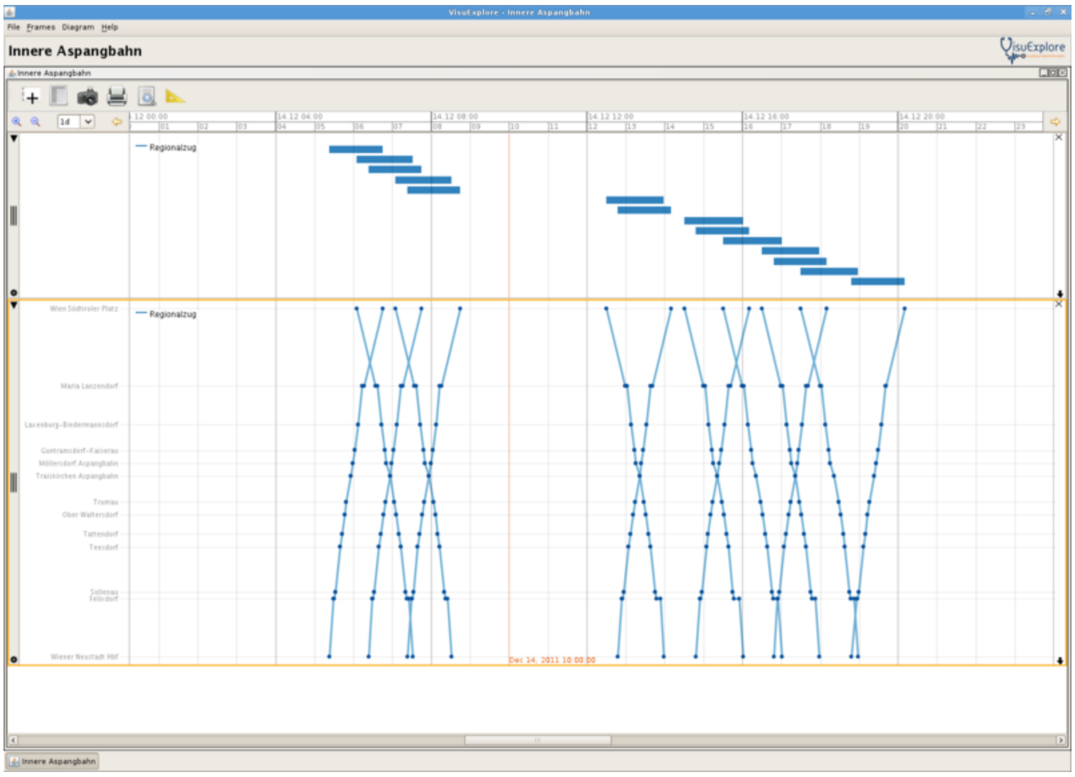
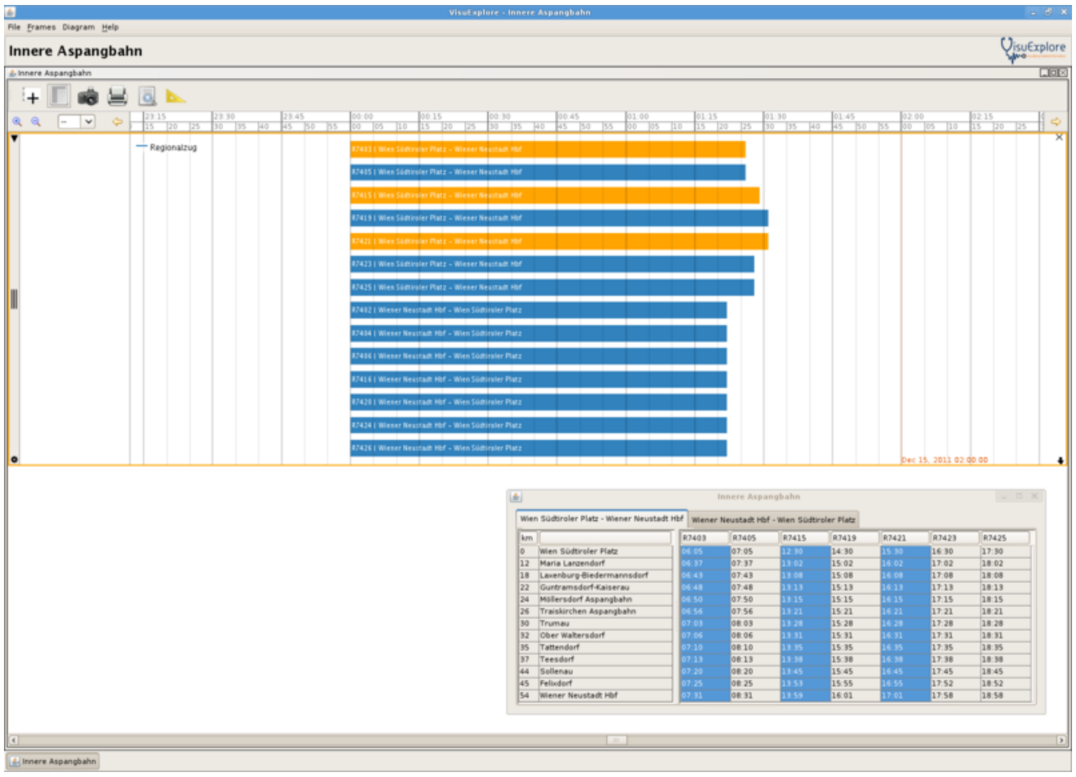


Figure 3: Visualizations supported by TrainVis 2


VisuExplore - Innere Aspangbahn
File Frames Diagram Help
Innere Aspangbahn
VisuExplore
Regionalzug
Dec 15, 2011 02:00:00
Wien Südtiroler Platz - Wiener Neustadt Hbf
Wiener Neustadt Hbf - Wien Südtiroler Platz

km		R7403	R7405	R7415	R7419	R7421	R7423	R7425
0	Wien Südtiroler Platz	06:05	07:05	12:30	14:30	15:30	16:30	17:30
12	Maria Lanzendorf	06:37	07:37	13:02	15:02	16:02	17:02	18:02
18	Laxenburg-Biedermannsdorf	06:43	07:43	13:08	15:08	16:08	17:08	18:08
22	Guntramsdorf-Kaiserau	06:48	07:48	13:13	15:13	16:13	17:13	18:13
24	Möllersdorf Aspangbahn	06:50	07:50	13:15	15:15	16:15	17:15	18:15
26	Traiskirchen Aspangbahn	06:56	07:56	13:21	15:21	16:21	17:21	18:21
30	Trumau	07:03	08:03	13:28	15:28	16:28	17:28	18:28
32	Ober Waltersdorf	07:06	08:06	13:31	15:31	16:31	17:31	18:31
35	Tattendorf	07:10	08:10	13:35	15:35	16:35	17:35	18:35
37	Teesdorf	07:13	08:13	13:38	15:38	16:38	17:38	18:38
44	Sollenau	07:20	08:20	13:45	15:45	16:45	17:45	18:45
45	Felixdorf	07:25	08:25	13:53	15:55	16:55	17:52	18:52
54	Wiener Neustadt Hbf	07:31	08:31	13:59	16:01	17:01	17:58	18:58

Figure 4: Using the data panel with the timetable bar chart

The textual timetable representation implemented in TrainVis was replaced by an interactive data panel which resembles the look of an actual timetable with a stations table on the left and a trains table on the right hand side. When a train table column is selected, the corresponding visual item is highlighted in the visualization (see Figures 4 and 5).

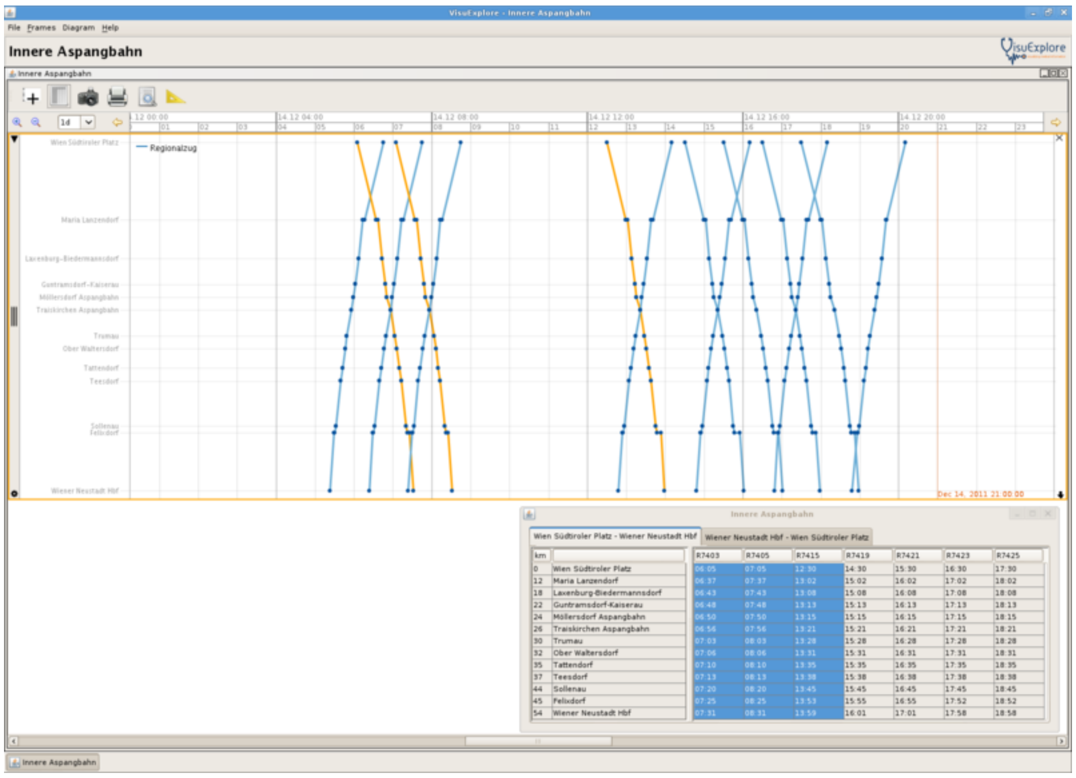
3.1 Timetable bar chart

In the timetable bar chart each train is represented by a bar spanning from its departure at the first stop to its arrival at the last stop. The following settings are available (see Figure 6):

Title Sets the title of the visualization which is displayed as a watermark.

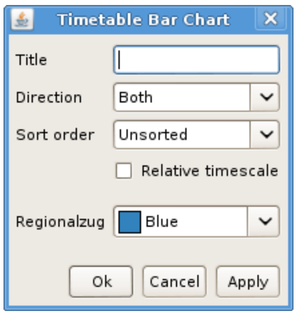
Direction Filters the trains based on their direction of travel.

Sort order Sorts the trains based on their direction of travel or displays them in the order they are specified in the input file.



km		R7403	R7405	R7415	R7419	R7421	R7423	R7425
0	Wien Südtiroler Platz	06:05	07:05	12:30	14:30	15:30	16:30	17:30
12	Maria Lanzendorf	06:37	07:37	13:02	15:02	16:02	17:02	18:02
18	Laxenburg-Biedermannsdorf	06:43	07:43	13:08	15:08	16:08	17:08	18:08
22	Guntramsdorf-Kaiserau	06:48	07:48	13:13	15:13	16:13	17:13	18:13
24	Möllersdorf Aspangbahn	06:50	07:50	13:15	15:15	16:15	17:15	18:15
26	Traiskirchen Aspangbahn	06:56	07:56	13:21	15:21	16:21	17:21	18:21
30	Trumau	07:03	08:03	13:28	15:28	16:28	17:28	18:28
32	Ober Waltersdorf	07:06	08:06	13:31	15:31	16:31	17:31	18:31
35	Tattendorf	07:10	08:10	13:35	15:35	16:35	17:35	18:35
37	Teesdorf	07:13	08:13	13:38	15:38	16:38	17:38	18:38
44	Sollenau	07:20	08:20	13:45	15:45	16:45	17:45	18:45
45	Felixdorf	07:25	08:25	13:53	15:55	16:55	17:52	18:52
54	Wiener Neustadt Hbf	07:31	08:31	13:59	16:01	17:01	17:58	18:58

Figure 5: Using the data panel with the Marey chart


Timetable Bar Chart

Title

Direction: Both

Sort order: Unsorted

☐ Relative timescale

Regionalzug: 🟦 Blue

Figure 6: Timetable bar chart settings dialog

Relative timescale Activates the relative timescale feature. The departure time at the first stop of a each train is mapped to 00:00 so that travel times can easily be compared.

Color choosers Dynamically created combo boxes allow for the assignment of a color to each train category present in the dataset.

3.2 Marey chart

Basically, a Marey chart [Mar75] is a time-distance diagram with the stations placed on the vertical axis proportional to their actual distance and time plotted on the horizontal axis. The following settings are available (see Figure 7):

Title Sets the title of the visualization which is displayed as a watermark.

Line width Sets the width of the lines representing the trains.

Point size Sets the size of the points representing the stops at a station.

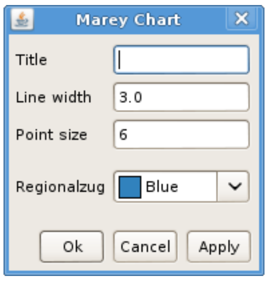
Color choosers Dynamically created combo boxes allow for the assignment of a color to each train category present in the dataset.

Running timetable queries on the visualized data is possible using the chart's query dialog which is accessible via a toolbar icon. The following options are available (see Figure 8):

From The desired departure station.

To The desired arrival station.

Date The desired travel date. Only dates in the current dataset's timetable period are selectable. In case the date field is left empty, every day in the timetable period is searched for matching connections.


Marey Chart

Title:

Line width: 3.0

Point size: 6

Regionalzug: 🟦 Blue

Figure 7: Marey chart settings dialog

Time The desired departure or arrival date depending on the selected option

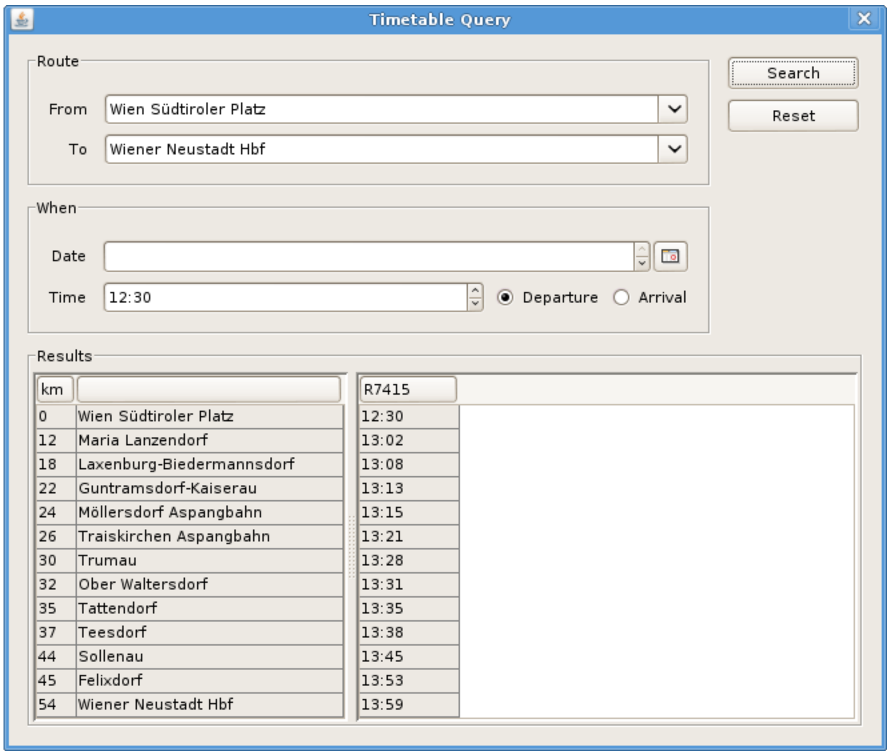
Search Starts the timetable query. Matching trains are displayed in the results panel. Just like with the timetable data panel, columns selected in the results panel are highlighted in the visualization (see Figure 9).

Reset Resets all entries to their default values and clears the results panel.

3.3 Input format

TrainVis 2 uses railML [NHSK04], an XML-based file format facilitating data exchange between heterogeneous railway applications, as its input format. The current railML specification contains subschemas for infrastructure, timetables, and rolling stock. TrainVis 2 does, however, only support a minimal subset expressive enough to specify the aspects relevant to the application, i. e., stations and their respective position along a track as well as trains and their category, operating period, and stops. Generally, if one aspect may be modeled in several redundant ways, only support for the machine-readable format is implemented.

A minimal railML example with a single train travelling from Station 1 to Station 2 is given in Listing 1. The infrastructure section uses *ocp* elements to define both stations and *crossSection* elements to assign them an absolute position along a track in meters. The timetable section starts with defining the timetable period and the available train operating periods. While both entities may include redundant human-readable definitions, only the *startDate* and *endDate* attributes of the *timetablePeriod* element as well as the *bitMask* attribute of the *operatingPeriod* element are evaluated. Next comes the definition of the available train categories. The rest of the timetable section deals with the available train parts and their composition. Each part is assigned a certain category and operating period as well as a list of stops. While every train may theoretically consist of several parts with different attributes, TrainVis 2 does only consider the first one and does not support special operating procedures like train coupling and sharing either.



Route

From: Wien Südtiroler Platz

To: Wiener Neustadt Hbf

When

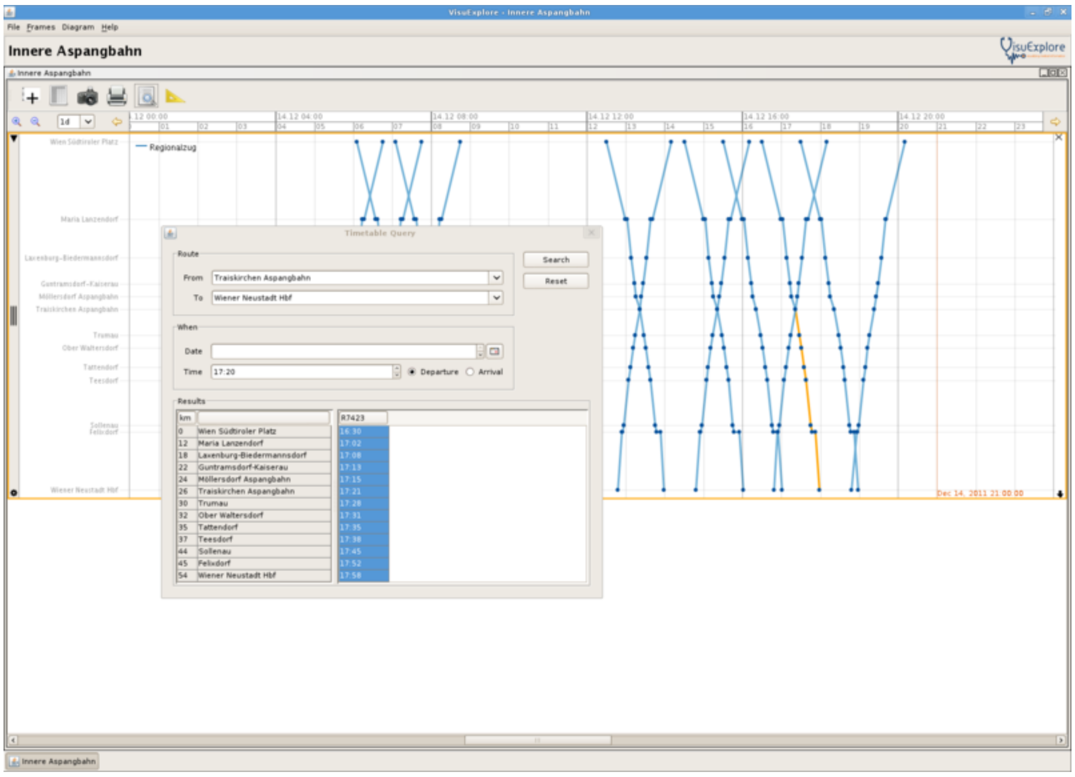
Date:

Time: 12:30 [✓ Departure] Arrival

Results

km		R7415
0	Wien Südtiroler Platz	12:30
12	Maria Lanzendorf	13:02
18	Laxenburg-Biedermannsdorf	13:08
22	Guntramsdorf-Kaiserau	13:13
24	Möllersdorf Aspangbahn	13:15
26	Traiskirchen Aspangbahn	13:21
30	Trumau	13:28
32	Ober Waltersdorf	13:31
35	Tattendorf	13:35
37	Teesdorf	13:38
44	Sollenau	13:45
45	Felixdorf	13:53
54	Wiener Neustadt Hbf	13:59

Figure 8: Marey chart query dialog



Route

From: Traiskirchen Aspangbahn

To: Wiener Neustadt Hbf

When

Date:

Time: 17:20 [✓ Departure] Arrival

Results

km		R7423
0	Wien Südtiroler Platz	16:30
12	Maria Lanzendorf	17:02
18	Laxenburg-Biedermannsdorf	17:08
22	Guntramsdorf-Kaiserau	17:13
24	Möllersdorf Aspangbahn	17:15
26	Traiskirchen Aspangbahn	17:21
30	Trumau	17:28
32	Ober Waltersdorf	17:31
35	Tattendorf	17:35
37	Teesdorf	17:38
44	Sollenau	17:45
45	Felixdorf	17:52
54	Wiener Neustadt Hbf	17:58

Figure 9: Querying the Marey chart

```xml
<?xml version="1.0" encoding="utf-8"?>
<railml version="2.1" [...]>
   <infrastructure id="[...]" name="[...]">
      <tracks>
         <track id="[...]">
            <trackTopology>
               [...]
               <crossSections>
                  <crossSection id="[...]" ocpRef="ocp_St1" pos="0"/>
                  <crossSection id="[...]" ocpRef="ocp_St2" pos="100000"/>
               </crossSections>
            </trackTopology>
         </track>
      </tracks>

      <operationControlPoints>
         <ocp id="ocp_St1" abbrevation="St1" name="Station 1"/>
         <ocp id="ocp_St2" abbrevation="St2" name="Station 2"/>
      </operationControlPoints>
   </infrastructure>

   <timetable id="[...]">
      <timetablePeriods>
         <timetablePeriod id="[...]" startDate="2011-12-11" endDate="2012-12-08"/>
      </timetablePeriods>

      <operatingPeriods>
         <operatingPeriod id="operatingPeriod_A" name="A" bitMask="0111110[...]"/>
      </operatingPeriods>

      <categories>
         <category id="category_R" name="R" description="regional train"/>
      </categories>

      <trainParts>
         <trainPart id="trainPart_R4711_1" categoryRef="category_R">
            <operatingPeriodRef ref="operatingPeriod_A"/>
            <ocpsTT>
               <ocpTT ocpRef="ocp_St1">
                  <times scope="scheduled" departure="06:00:00"/>
               </ocpTT>
               <ocpTT ocpRef="ocp_St2">
                  <times scope="scheduled" arrival="07:00:00"/>
               </ocpTT>
            </ocpsTT>
         </trainPart>
      </trainParts>

      <trains>
         <train id="train_R4711" trainNumber="R4711" type="commercial">
            <trainPartSequence sequence="1">
               <trainPartRef ref="trainPart_R4711_1"/>
            </trainPartSequence>
         </train>
      </trains>
   </timetable>
</railml>
```

Listing 1: A minimal railML example

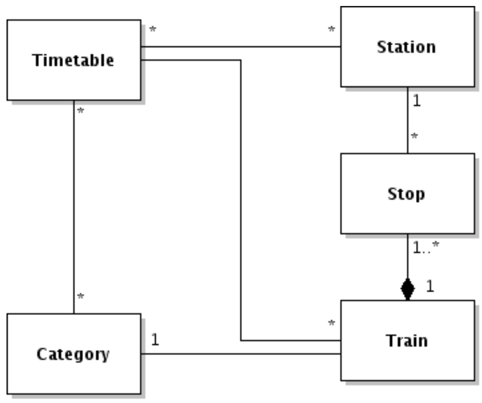


Figure 10: TrainVis 2 model classes

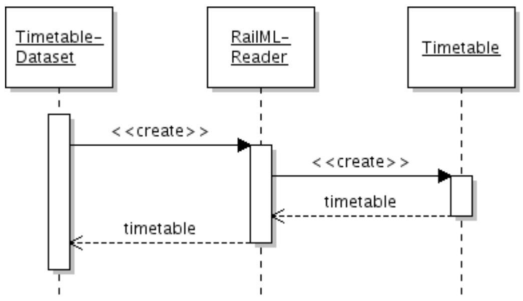


Figure 11: Creation of a timetable object

3.4 Involved classes

The subset of relevant information extracted from the railML file is straightforwardly modeled by aptly named classes (Figure 10). The prefuse data structures used by the two new visualizations *TimetableBarChart* and *MareyChart* are created by the *TimetableDataset* object which is backed by a *Timetable* object originating from a *RailMLReader* instance (Figure 11).

Like all visualizations provided by VisuExplore, both visualizations added in the course of the TrainVis 2 project consist of a respective *Visualization*, *SettingsDialog*, and *Settings* class where the latter acts as a bridge between the former two and manages the persistable attributes. It may hold references to objects like the currently active *Direction* or *SortOrder* object used in the process of filtering and querying the timetable dataset. These objects are created according to the user's preferences by the *SettingsDialog* and ultimately processed by prefuse *Actions* in the *Visualization* object.

References

[Hee04] Jeffrey Heer. prefuse: a software framework for interactive information visualization. Master's thesis, University of California, Berkeley, December 2004.

[Mar75] Étienne-Jules Marey. La méthode graphique dans les sciences expérimentales (Suite). In *Physiologie Expérimentale*. G. Masson, 1875.

[MPP07] Markus Martin, Hansjörg Pripamer, and Georg Prohaska. TrainVis. http://ieg.ifs.tuwien.ac.at/projects/trainVis/, 2007.

[NHSK04] Andrew Nash, Daniel Hürlimann, Jörg Schütte, and Vasco Paul Krauss. RailML – a standard data interface for railroad applications. In *Computers in Railways IX*, pages 233–240, Dresden, Germany, May 2004.

[RMA+10] Alexander Rind, Silvia Miksch, Wolfgang Aigner, Thomas Turic, and Margit Pohl. VisuExplore: Gaining new medical insights from visual exploration. In *Proceedings of the 1st International Workshop on Interactive Systems in Healthcare (WISH 2010)*, pages 149–152, Atlanta, Georgia, United States, April 2010.

[Wei07] Peter Weishapl. TimeVis: Visualizing temporal data using prefuse. Bachelor's thesis, Vienna University of Technology, March 2007.