Centralizing and Migrating Operational Infrastructure Databases

Alexander Kuckelberg [a,1], Bianca Multykin [a,2]
[a] VIA Consulting & Development GmbH
Römerstr. 50, 52064 Aachen, Germany
[1] E-mail: a.kuckelberg@via-con.de, Phone: +49 (0) 241 463662 16
[2] E-mail: b.multykin@via-con.de, Phone: +49 (0) 241 463662 39

Abstract

Since years and decades, IT systems are used to plan, to monitor and to control train operations and railway traffic on network regions. Especially in technically advanced railway networks, the usage of computer based systems for dispatching and controlling traffic started quite early, e.g. in the 90s. This implies the necessity to update and renew outdated structures nowadays.

As IT system performance, database sizes and functionalities grew within the last decades, a wide range of existing system limitations are not valid anymore and can be overcome by new systems, processes and hardware. Larger data sets and therefore the migration and aggregation of valid data sets within one new, larger data set are possible now.

However, for operational systems it is highly advisable to follow an evolving strategy for the migration of distributed structures instead of a revolutionary approach to ensure the operability of working systems and ongoing operations. Such a strategy requires the migration of existing data sets and processes whereby the question arises how to migrate e.g. formerly overlapping infrastructure areas, how to aggregate semantically identical data sets with distinct technical keys etc.

This paper introduces these challenges from various points of view and presents approaches chosen by the authors to establish such a migration process of existing and running operational infrastructure databases. It focuses on technical aspects to migrate and aggregate infrastructure data but also outlines challenges with respect to migration of workflows and processes towards centralized services.

Keywords
Infrastructure Topology, Databases, Data Consolidation, Legacy Systems

1 Introduction

Current operational IT systems used for train control and dispatching often realize a microscopic infrastructure model as a base data model. The migration and aggregation of legacy systems into new and larger systems arises several challenges, e.g. consolidation of different, probably overlapping infrastructure data sets, migration problems with respect to unsynchronized data maintenance and guarantee of consistency for resulting consolidated data.

Within this chapter, two elementary aspects of the problem are introduced: The microscopic infrastructure data model used by legacy systems and the problem of overlapping responsibilities and unsynchronized microscopic data sets.

After introducing the basic data structures and clarifying the problems implied by distributed IT systems, the resulting challenges are described within Chapter 2 for some selected aspects in detail. Chapter 3 outlines approaches which have been selected (and implemented) to solve the challenges and finally Chapter 4 concludes.

1.1 Microscopic Infrastructure Model

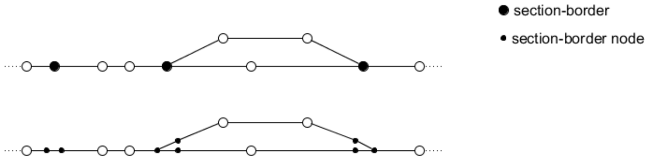


Figure 1: Mesoscopic (above) and microscopic (below) view on railway topology.

In railway operations research approaches as well as in operational systems the modelling of railway infrastructure is an essential step towards a functional data model that can be processed automatically by IT systems. The granularity of such models determines the capability of systems set upon these models. For operational systems like train control and dispatching, microscopic infrastructure models turned out to fit functional requirements in a good manner. The systems considered by this work implement microscopic infrastructure models as follows:

- The real infrastructure is modelled by a graph model consisting of nodes and edges, where nodes represent infrastructure elements (signals, stopping positions, switches and crossings, axle counters etc.) and edges represent tracks connecting the infrastructure elements.
- Track sections – tracks without branches – are represented by a sequence of inner nodes (with exactly two neighbours), bordered by two outer nodes.
- Outer nodes and therefore section-border nodes are track end nodes, buffer nodes, branches of switches and crossings, transition nodes towards new logical/operational node affiliation, etc. (Figure 1). Neighboured sections are connected by an edge.
- All nodes are logically clustered into operational control points (OCP), and sections end when entering a new operational control point. Consequently, these sections end with an OCP bordering node and are linked to the corresponding OCP bordering node starting a section of the other OCP.
- All nodes have a mileage value, ordering section nodes within a section in a monotonous manner. Consequently, all nodes of one section are ordered and the section itself gets an implied direction.
- Switches are modelled by three, crossings by four section-border nodes (one for each branch); interconnections between these nodes represent the possible routing throughout a switch or crossing, resp. (Figure 2).

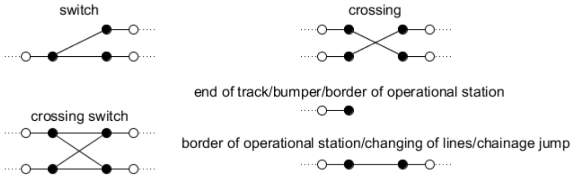


Figure 2: Modelling of interconnections between sections.

Upon these microscopic nodes technically secured routes are defined (Figure 3). These routes consist of tracks of a station or a line. In theory a route is a path in the graph consisting, inter alia, of a start and an end node, and the course of the route throughout the graph (defined by branching information for each switch passed by). With routes, the interlocking behaviour and dependency by signals is modelled.

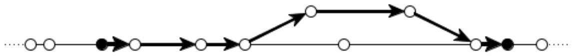


Figure 3: A route defined as a path in the graph, starting with an entry signal.

1.2 Distributed Structures and Data Sets

In this work, we consider the railway infrastructure of a large area, e.g. a whole country. We assume that the infrastructure data is distributed into regions, where each region is controlled by its responsible operational control centre. Every control centre has the infrastructure data of its own region and a small glimpse of the infrastructure data across its border to model the cross-border coherences. So the given infrastructure data of every control centre is its region internal infrastructure data extended by its own data of the infrastructure across its border. It is not ensured that the overlapping infrastructure data of two operational centres is synchronized, as every control centre only takes care of its own database. As the data sets are maintained independently the possibility of historically grown apart data in border regions is given. The infrastructure data of each control centre is considered to be consistent.

Figure 4: Regional data responsibility and data sets with overlapping region borders.

In addition to regional oriented infrastructure data, every control centre has its own content oriented data like train types, public holidays, braking tables etc. This data should be similar for every control centre, but e.g. deviation in naming might exist and some control centres may have more extensive data sets than others or just data that is not relevant for others as a public holiday that is only regional.

2 Challenges and Migration Problems

Data sets of operational control centres – as representatives of regions – follow certain semantics: All centres have similar data, acting as master data of more or less static conditions that form a certain common data basis. This data set is considered as content oriented data and has to be treated in another way than region oriented data, where the maintenance responsibility can be clearly assigned to one control centre (except for cross-border coherences). The characteristics of these two principle data sets are described in the following sections. The handling when migrating depends on these data characters.

2.1 Content Oriented Data Sets

The content oriented data of an operational control centre contains all information considered as "static dictionary data", often called master or framework data. It includes for instance the list of train types, braking tables (braking percentage/LZB braking curve/braking delay/ETCS), referenced keys and data of tractive units. Content oriented data which exists in various regions should be equal in every regions data set to ensure a common perspective, e.g. a unique locomotive number should identify the same engine in all regions.

So an aggregating data set would secure the consistency of the perspectives of the regions. Even though there is data which may not be relevant for every region, a universal database for all regions would bring advantage as it is of importance for the optimization of cross-border process flows. A public holiday which does not exist in one region but in its adjacent region is important, as it may influence its workload or its timetable as example.

The main challenges for content oriented data sets is to determine how "new data sets" fit to the existing data sets, to detect content changes for the same data entries and to select the correct shaping of a data entry.

2.2 Regional Data Sets

Regional data sets considered within this paper consist primarily of infrastructure data following the detailed microscopic infrastructure data model (Chapter 1.1). One characteristic of regional data sets is that they can be assigned to a region uniquely, so exactly one operational control centre is responsible for its regions allocated subgraph.

Regional data sets might overlap at their bordering areas. This means, that for operational reasons a regional data set might contain data of topology and graph areas which are not in its operational control centres responsibility and vice versa.

While infrastructure data within a region is expected to be consistent and consistency is ensured by the legacy systems itself, migrated databases have to deal with bordering areas, where information from affected regions might contradict each other.

Additionally, not only the topology and infrastructure of two regions might be contradicting within the bordering area, but also the elements and information contained within sections of the bordering area, e.g. distances between nodes (e.g. distance signal to main signal), braking distances, maximum speeds, gradients, train control equipment etc.

2.3 Synchronization and Collaboration, Process Flows

One of the central problems when migrating similar and complementary data sets in the context of databases is the identity of data entries. Region identifiers of data entries might not be unique within the more global context of centralized and migrated databases anymore.

In other words, it should be possible to load different content versions of one data entry and to manage all its occurrences.

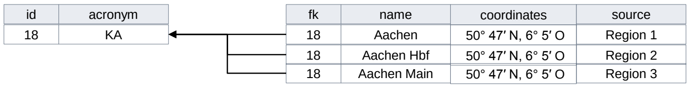
Another important aspect when migrating legacy databases are existing workflows. As mentioned, the migration should follow an evolving approach which directly implies, that legacy workflows remain similar and only change stepwise.

Therefore, migrating regional data sets also includes migrating workflows, e.g. the frequency of data version publication and propagation for each region and how new versions are integrated into the migrated database. Two principle approaches are possible:

- Direct reaction whenever new data versions are published by a region or
- Implementation of aggregation, enforcement of new workflows e.g. collection and propagation periods.

3 Centralizing and Migrating

In our approach every data delivery of a region is stored as one version. Data is transferred into an object-identity set and a shaping set, called splitted schema. The object-identity set contains the technical global keys of the new enlarged data set and columns forming semantical keys. Semantical keys are derived from regional data sets and remain equal in every regions delivery. So these keys identify entries throughout all regions, e.g. the combination of a tractive unit series number and a company identifier for a tractive unit (as only one of these attributes would not identify it uniquely). The shaping data set contains the remaining data content of every entry and references its object identity as outlined by Figure 5.



id	acronym
18	KA

fk	name	coordinates	source
18	Aachen	50° 47′ N, 6° 5′ O	Region 1
18	Aachen Hbf	50° 47′ N, 6° 5′ O	Region 2
18	Aachen Main	50° 47′ N, 6° 5′ O	Region 3

Figure 5: Separation of data entry identity from entry content (example OCP).

The object-identity is generated once for each semantical key, the shaping set grows with every delivery and is associated with the corresponding version.

Content oriented data from different regions is merged on behalf of common object identity and multiple shapings (Chapter 3.1). Region oriented data is restricted to entities related to the merged region (Chapter 3.2) therefore a methodology to ensure consistency at borders is developed (Chapter 3.3).

For both data types object dictionaries are introduced to map semantical keys of incoming entities to objects representing the target entities with their associated attributes. These object dictionaries represent the current state of the merged target data set as they get extended with every occurrence of a new semantical key while merging. Moreover, an entity-wise merging in a hierarchical order, derived from data entity dependencies, is implemented to ensure hierarchical data entity references.

3.1 Merge of Content Oriented Data Sets

For the merge of content oriented data sets the already merged content oriented data is loaded from the target data set and added to the dictionaries identified by their semantical keys. In this way, the information about already existing and known content oriented data is provided for further merging.

On the other side, delivered data has to be merged into the target data sets. First the imported data is converted into the splitted target scheme. Every splitted data entity gets saved into its object identity and shaping data set. While merging a data set, it is iterated pairwise through these sets for each delivered entity. For every considered pair the existence of its semantical key in the object dictionary is verified and depending on its occurrence it is acted (Figure 6).

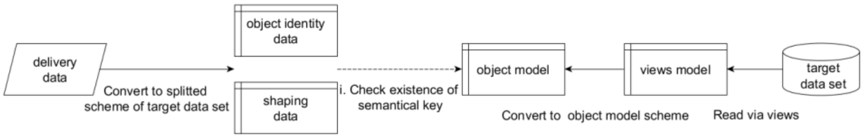


Figure 6: Merging components for content oriented data sets.

During the merge process the object dictionaries represent the current state of the merged target data set. By merging and saving into the target data set the local technical keys of the delivery data set are replaced recursively by the technical keys of the target data set on behalf of the dictionaries. The hierarchical merging enables correct re-referencing to the keys of earlier merged data entities (references which are part of the semantical key included). So with every merge-step first the re-referencing takes place.

In the next step, the existence of the entities identity – the semantical key – within the corresponding dictionary is checked.

If the identity is missing, the object identity is added to the target data set and the object identity of the entities splitted data gets updated by the new generated technical key. Additionally the object repository gets extended by the persisted entity.

If the identity already exists the entities object identity key is replaced by the mapped one of the target data set.

In both cases, the shaping data entity component is added to the target data set and associated to the updated object identity as well as to the version defining the data entities validity (Figure 7).

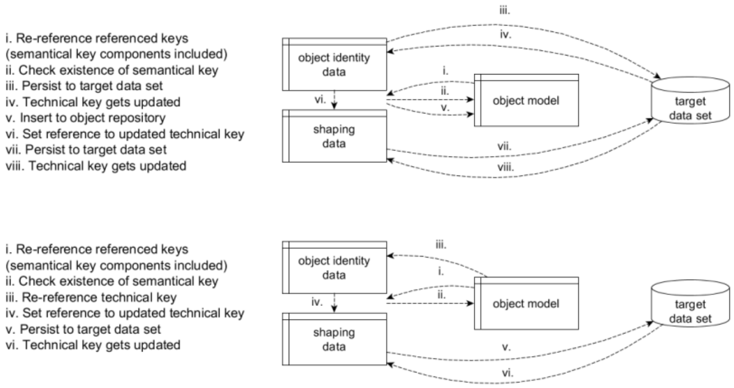


Figure 7: Merge of content oriented base data sets.
Above: Workflow for a new data entity.
Below: Workflow for an already existing data entity.

3.2 Merge of Region Oriented Data Sets

Region oriented data sets are merged in a different way than content oriented data sets: The regions infrastructure data of adjacent regions is excluded for every regions delivered data set. In consequence problems might occur while merging, if infrastructure data which belongs to the merged region references infrastructure data of another region. In this case, references are modelled by semantical keys which get saved as a substitute for later integration.

Again, like in the merge of the content oriented data, the delivered data entities get transferred into the splitted schema of the target data set due to the merge (object identity and shaping data). For region oriented data in turn the object identity data only contains the mapped technical keys of the target data set and the technical keys of its delivery data set.

As the amount of infrastructure data usually is very huge, key-maps get introduced which map the technical keys of a regions region oriented delivery data sets from earlier merges to the technical keys of the target data set. These resulting key-maps are loaded from the target data set as a first, preparing step.

Non-existing technical keys of the regions delivery data set in the key-map imply persistence of the object identity data to the target data set and hereby include the generation of new technical keys. By every new generation, the key-map gets extended. If the technical key already exists within the key-map, mapping can be performed directly.

In this way for all region oriented data, merging can be limited to key-map activities. Complete re-referencing is supported and the existing (shaping) infrastructure data of the target data set is not required. It can remain within the target data set without any access.

The content oriented data set is merged before the regional oriented data set. Due to this hierarchical treatment in the current state of the merge the object repository contains all content oriented data within the corresponding dictionaries for further usage. So re-referencing of referenced content oriented data can further on take place through the object repository.

With every merge-step the object repositories get extended by the re-referenced region oriented data entity as well, whereby in this case only for test purposes.

Independent from the existence of the data entities technical key in the key-map, the entities shaping data component is added to the target data set and associated to the object identity as well as a version defining the data entities validity (Figure 8).

Additionally, operation control points and lines are expected to be valid for all regions and are treated as overall data with given and predefined identities. Delivered data sets will only extend these data entities in the target data set with corresponding warnings.

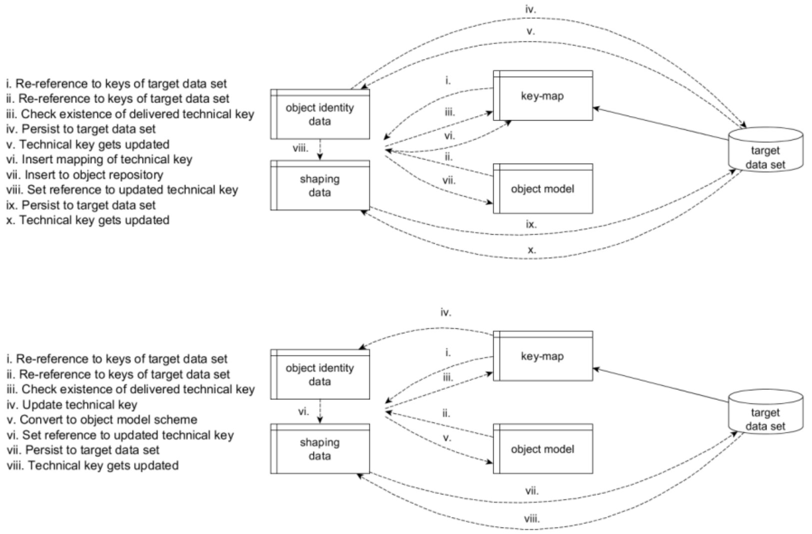


Figure 8: Merge of region oriented data sets.
Above: Workflow for a new data entity.
Below: Workflow for an already existing data entity.

3.3 Border Analysis of Region Oriented Data Sets

As described in Chapter 3.2., infrastructure data of all neighbour regions is filtered. When merging regional data, the bordering areas might become inconsistent, e.g. missing tracks within the neighbouring region. Border consistency is verified by a new, supporting data entity, the connectors. Connectors are elements which conclude information about the border conditions of the regions potential border crossings, identifying a position within the graph semantically. Two shaping data sets (border-node and border-route connectors) concretise infrastructure and route specific information at connector positions.

Bordering Infrastructure

Border-node connectors are introduced to represent infrastructure specific border information. A border-node connector references an associated border-node of the region and contains indicator values that should fit to the indicator values of an associated border-node connector belonging to the adjacent neighbour region. Indicator values are e.g. distances between the border-node and closest signals (distant and main) along the track, current braking distances or track characteristics like gradient and curve. Some values moreover are computed for inbound or outbound trains separately (e.g. closest signals). Connectors are identified while merging the regional data sets due to semantical information and represent locations within the graph (border-nodes or track ends), where regions might join and where border consistency has to be checked.

After merging all region oriented data, a newly developed graph iteration algorithm determines the indicator values of all identified border-node connectors and assigns the values to them. The iteration determines distances considering mileage changes or mileage direction changes. Break conditions for the graph iteration are e.g. the reach of graph borders, exceeding of a defined maximum distance or the successful determination of all values. Figure 9 outlines this algorithm.

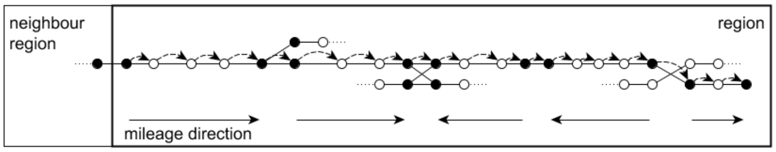


Figure 9: Graph iteration algorithm for indicator value determination of a border-node connector.

Bordering Routes

Cross-border routes have to be considered in the border analysis as well. Routes have to be split up at borders if they are belonging to different regions. With a region perspective the inner route segment might be an inbound or outbound segment starting or ending at a border-node referenced by a border-node connector. Border-route connectors are generated for these cross-border routes and semantical keys are derived, so matching route segments can be associated to each other at data retrieval time.

As for border-node connectors border-route connectors are enriched by route indicator values. These indicators contain e.g. the partial routing information. Figure 10 illustrates the algorithm which iterates throughout the route by its course and identifies the inner route course of the region. In dependence from the routes direction, the algorithm starts at the end- or the start node of the route, as regional data across the border was filtered, until it reaches the border-node of the associated connector.

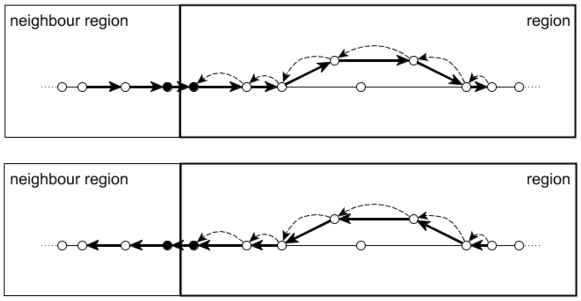


Figure 10: Graph iteration algorithm for indicator value determination of border-route connector.

Data Consistency

Data consistency checks for data of neighbouring regions are performed on behalf of border-node connectors and border-route connectors by their existence and their indicator values.

A first topology check determines the existence of fitting border-node connectors for adjacent regions (border-node to border-node or track-end without partner). If the topology of bordering nodes is consistent the next steps will evaluate indicator values with respect to infrastructure and routes.

Border-node connectors are evaluated with respect to reasonable indicator value matches, e.g. distance of cross-bordering distant and main signals, consistent train protection, tunnel cross-section, gradient and curve value consistency and more (Figure 11). Route-connectors are e.g. evaluated with respect to complementary partial routing.

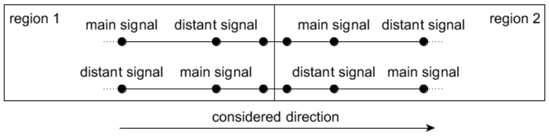


Figure 11: Consistency check example (signal positioning & border-node connectors).

4 Conclusion

This paper presented an approach to migrate and consolidate existing legacy infrastructure databases. With the approaches it becomes possible to centralize existing infrastructure data sets currently used in several operational IT systems with microscopic data models.

The approach therefore allows bringing together distributed and currently independent data sources and setting up new, centralized functionalities on top of new and enlarged data models.

Moreover, the paper outlines less technical aspects when migrating and implementing such IT systems. Anyway, final experiences and evaluations will not be possible until the new IT systems are in operation, the algorithms will be adopted and modified accordingly while implementing and evolving the system.

References

Kuckelberg, A., 2015. *Graph Databases and Railway Operations Research Requirements*, EDBT/ICDT Workshops, 183-188.

Kuckelberg, A., Seybold, B., 2013. *Adaptive Rule-Based Infrastructure Modelling*, Proc. of the 5th International Seminar on Railway Operations Modelling and Analysis, Copenhagen.

Hundt, R., 2005. *Betriebszentralen der DB Netz AG – Die Revolution in der Betriebsführung*.Signal + Draht. Vol 97, Nr. 1, ISSN 0037-4997, S. 3.

Hirschlach, J., 2016, *Aufbruch ins Computerzeitalter* Eisenbahn-Magazin. Nr. 3, ISSN 0342-1902, S. 45.

Bormet, J., Rausch, R., 2017, *Weichen stellen für die Zukunft der Betriebssteuerung*. Deine Bahn. Bahn Fachverlag GmbH