Versionierung von Infrastrukturdaten und Verfügbarkeiten

Infrastrukturdaten wie Topologie, Sicherungstechnik und abgeleitete Nutzungsmöglichkeiten sowie zeitliche Ausprägungen sind zentrale Grundlage verschiedener Systeme des Bahnbetriebs, z. B. zur Disposition, Zugmeldung, Fahr- und Baubetriebsplanung, Analyse und Statistikerhebung. Dieser Artikel beschreibt Konzept und Modellierung eines Systems zur versionierten Infrastrukturverwaltung.

Einleitung

Leit- und Betriebssysteme des Bahnbetriebs wie Dispositionssysteme, Systeme der Zugstandortverfolgung, Zugsicherungs- und Zugsteuerungssysteme sowie Planungssysteme z. B. zur Fahr- und Baubetriebsplanung basieren auf Infrastrukturdaten wie Gleistopologien, Nutzungsmöglichkeiten aufgrund verwendeter Stellwerks- und Sicherungstechnik, die aufgrund von Baumaßnahmen, Neubauten und Anpassungen von Bestandssystemen dynamischer Natur sind und sich im Lauf z. B. eines Fahrplanjahres punktuell ändern können. Ergänzt werden solche infrastrukturbezogenen Daten in der Regel um notwendige Stamm- oder Basisdaten wie Zuggattungen, Linienkennungen, Triebfahrzeugverzeichnisse etc. sowie verkehrliche und betriebliche Informationen (z. B. Anschluss- oder Wartezeiten), organisatorische Daten wie Strecken oder Betriebsstellen, die ebenfalls regelmäßigen Anpassungen unterliegen.

Infrastruktur- und Basisdatenanpassungen erfolgen kontinuierlich bzw. zu exponierten Zeitpunkten wie einem Fahrplanwechsel, finden aber auch unterjährig und ggf. bedingt durch kurzfristige und damit nicht planbare Ereignisse statt, was einer statischen Datengrundlage für einen festen und planbaren Zeithorizont wie z. B. ein Fahrplanjahr widerspricht und Systeme erfordert, die diese Änderungen verwalten und anderen Systemen der Dynamik entsprechend aufbereiten.

Im Rahmen des Projekts PRISMA (Projekt Re-Design Informationssysteme Betrieb und Modernisierung der Architektur) der DB Netz AG haben sich diese Anforderungen in sehr spezieller Form gestellt und eine spezifische Umsetzung für die von den Autoren mit entwickelte und später noch grob skizzierte Systemkomponente LeiDa-D (Leitsystem Datenhaltung – Datenhaltung) erfordert. Davon ausgehend erfolgte eine Weiterentwicklung und allgemeingültigere Formulierung der Versionierung von Daten sowie dies unterstützender Systeme, z. B. der Baubetriebsplanung oder Studien zu Alternativtopologien. Die besondere Herausforderung sind vor allem strukturelle Infrastrukturänderungen, d. h. Ereignisse, die die Topologie verändern und damit die größten Auswirkungen auf nachfolgende Systeme vermuten lassen.

In diesem Artikel werden grundlegende Konzepte der Versionierung sowie des daraus resultierenden Datenmodells beschrieben, bevor konkrete Realisierungsaspekte des generischen Ansatzes vorgestellt werden. Bild 1 illustriert zeitliche und örtliche Herausforderungen einer versionierenden Datenhaltung.

Versionierung und Versionsvektoren

Die Versionierung der betrachteten Daten folgt verschiedenen Leitgedanken bzw. Konzepten: Gruppierung von Daten nach Datenbeständen bzw. Fachobjekten, eine eindeutige Eigentümerschaft jedes Datums, eine kontinuierliche Weiterentwicklung des Datenbestandes durch dessen Eigentümer sowie verschiedener Status.

Sei daher $E=\{e_1,\ldots\}$ eine Menge von Eigentümern, $S=\{s_1,\ldots\}$ eine Menge von Status sowie Z der Wertebereich gültiger Zeitpunkte und $Z_\perp=Z\cup\perp$ dieselbe Menge erweitert um einen Wert für „undefinierter Zeitpunkt". Die Menge E definiert die Verantwortung für Datenbestände und Datensätze, Statuswerte können z. B. „vorläufig", „freigegeben", „zurückgenommen" o. Ä. sein. Das Konzept ist auch valide, wenn das versionierende System selbst Eigentümer von Daten ist.

Mit dieser Grundlage ist eine Version v (eines Datenbestands/eines Fachobjekts) definiert als Tupel $v=(e,s,z^{von},z^{bis})\in E\times S\times Z\times Z_\perp$. Der Zeitpunkt z^{bis} des Gültigkeitsendes darf im Gegensatz zum Zeitpunkt z^{von} undefiniert sein. Alle existierenden Versionen bilden die Menge $V=\{v_1,\ldots\}$. Eine Version beschreibt Status und Gültigkeit eines Eigentümers. Jeder Datensatz im System wird genau einer Version zugeordnet.

Dr. Alexander Kuckelberg
Geschäftsführer, Softwareentwicklung, Architektur und Modellierung
VIA Consulting & Development GmbH, Aachen
a.kuckelberg@via-con.de

Mario Trappein
Softwareentwicklung, Prozesse und Qualitätsmanagement
VIA Consulting & Development GmbH, Aachen
m.trappein@via-con.de

Homepageveröffentlichung unbefristet genehmigt für VIA Consulting & Development GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt / © DVV Media Group GmbH

Es ergibt sich des Weiteren die Menge der zu einem Zeitpunkt zϵZ gültigen Versionen als Versionsvektor $V_z=\{v_{z,1}, v_{z,2},\ldots\} \subseteq V$, bei dem für jedes $v_{z,i}=(e_{z,i},s_{z,i},z^{von}_{z,i},z^{bis}_{z,i}) \in V_z$ und einer akzeptierten Statusmenge S={s_1,…} gilt: $s_{z,i}$ϵS und $z^{von}_{z,i} \le z < z^{bis}_{z,i}$[1] und es gibt keine andere Version desselben Eigentümers und zeitlicher Bedingung mit späterem Gültigkeitsbeginn. Der Versionsvektor beschreibt damit die zum angefragten Zeitpunkt jeweils gültigen Versionen und den damit verknüpften Datenbeständen. Bild 2 illustriert Aspekte des beschriebenen Konzepts.

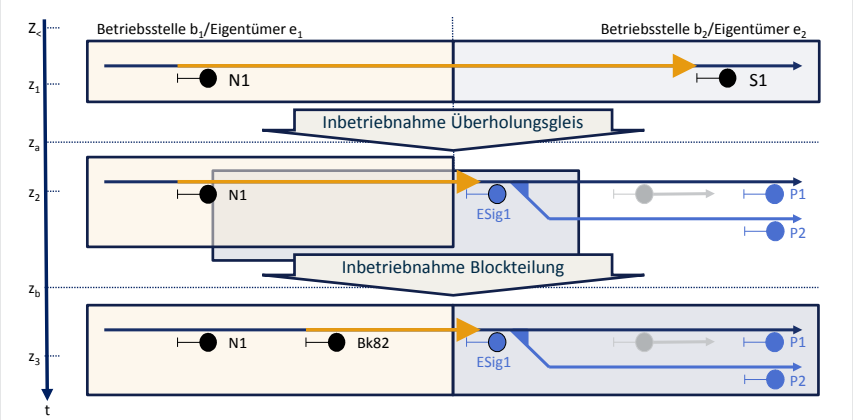
Life-Cycle-Management von Infrastruktur- und Basisdaten

Die Bereitstellung von neuen Datenbeständen mindestens eines Eigentümers bedeutet für einen Gesamtdatenbestand, dass sich ein Teil ab einem Zeitpunkt bzw. für einen Zeitraum ändert und diese Änderungen auch Daten unterschiedlichen Reifegrades – abgebildet durch die zuvor eingeführten Status – beschreiben können.

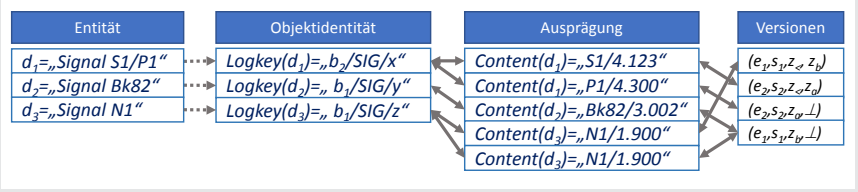
Es wird davon ausgegangen, dass die Bereitstellung eines Datenbestands durch einen Nutzer erfolgt, der gleichzeitig als Eigentümer von Daten des Datenbestands verstanden werden kann, nicht aber für alle Daten gelten muss. Dies ist z. B. der Fall, wenn Vorschau und Vormeldebereiche in Regionen anderer Eigentümer/Nutzer hinein (z. B. zur Anzeige zulaufender Zugfahrten) Teil des bereitgestellten Datenbestands sind. Der Bereitsteller eϵE eines Datenbestandes muss daher nicht Eigentümer aller enthaltenen Datensätze sein.

Weiterhin wird eine Menge F={f_1,…} von Fachobjektklassen (vereinfacht auch Fachobjekt/FO) eingeführt. Fachobjekte können regionale Infrastrukturelemente, Bereiche oder Fahrstraßen sein, aber auch Basisdaten. Jeder Datensatz gehört genau einem Fachobjekt an.

Eine Menge dat=(e, s, z^{von}, z^{bis}, DS) wird als Datenbereitstellung bezeichnet, mit Eigentümer e, Status s, Gültigkeit $(z^{von}, z^{bis}) \in Z \times Z_\perp$ und einer Menge DS={d_1, …, d_t} von Datensätzen, die die eigentlichen Inhaltsdaten der Bereitstellung darstellen. Für alle Datensätze dϵDS sind die Relationen fo(d)ϵF und owner(d, e)ϵE definiert; fo gibt das Fachobjekt, owner den Eigentümer eines Datensatz dϵDS an.



1: Umbau eines Bahnhofs und Umrüstung der Blockteilung: Versetzen des Signals S1 samt Umbenennung, Inbetriebnahme Überholungsgleis, Neuerrichtung des Einfahrsignals, Anpassung der Blockteilung, Definition neuer Fahrstraßen. Beide Infrastrukturstände sind versioniert abrufbar. Werden die Betriebsstellen zudem von unterschiedlichen Eigentümern gepflegt, sind weitere konsistenzerhaltende Maßnahmen wie Prüfung von Topologie und Fahrwegkonsistenzen zwischen den diversen Versionen notwendig



2: Versionen $v_{1,1}=(e_1,s_1,z_<,z_b)$, $v_{2,1}=(e_2,s_2,z_<,z_a)$, $v_{2,2}=(e_2,s_2,z_a,\perp)$ und $v_{1,2}=(e_1,s_1,z_b,\perp)$ die mit Datenbeständen des Eigentümers e_1 sowie e_2 verknüpft sind (s_1, s_2 beliebige Status). Für Anfragezeitpunkte z_1, z_2 und z_3 ergeben sich die Versionsvektoren $V_{z1}=\{v_{1,1}, v_{2,1}\}$, $V_{z2}=\{v_{1,1}, v_{2,2}\}$ und $V_{z3}=\{v_{1,2}, v_{2,2}\}$ und den diesen Versionen zugeordnete Datensätze

Zu einer Bereitstellung dat=(e, s, z^{von}, z^{bis}, DS) gehört eine neue Version v=(e, s, z^{von}, z^{bis}). Ein Datensatz dϵDS der Bereitstellung wird übernommen und mit v verknüpft, wenn owner(d, e)=e gilt. Des weiteren bezeichnet DS^f={dϵDS| fo(d)=f} die Datensätze eine Fachobjekts fϵF.

Beispiele für DS^f können je nach Modellierung die Menge aller Signale, Spurplanknoten, Zuggattungen oder Fahrstraßen sein. Für das Signal „ESig1“ des begleitenden Beispiels (Bild 1) gilt zudem owner(„ESig1“, e_1)=e_2 und owner(„ESig1“, e_2)=e_2, d. h. nur wenn „ESig1“ von e_2 in einer Datenbereitstellung gestellt wird, wird der entsprechende Datensatz übernommen, womit mehrfache Ausprägungen einer Entität verhindert werden.

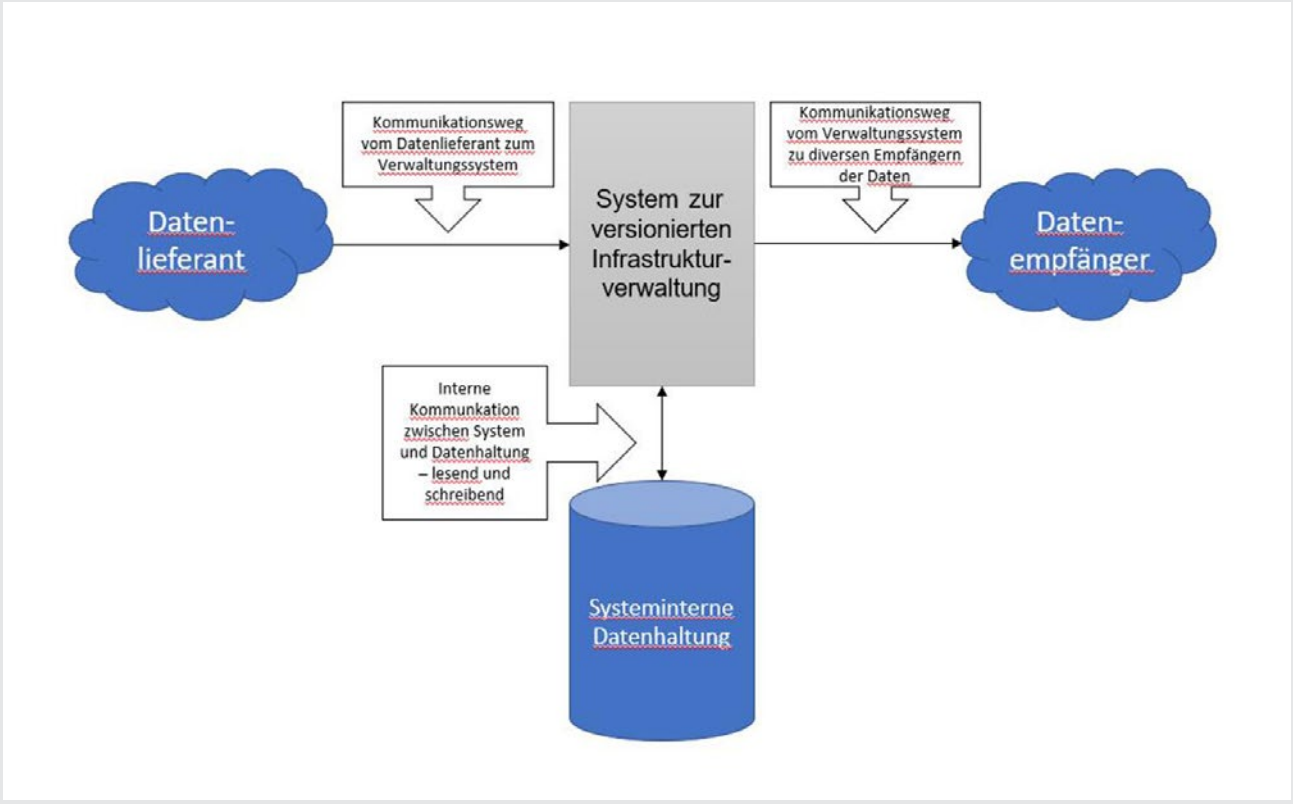
Zur Übernahme eines Datensatzes dϵDS ist dieser in einen identifizierenden Datensatzanteil mit logischem, unveränderlichem und daher nicht versionierten Schlüssel logkey(d) („Objektidentität“) sowie einen veränderlichen, versionierten Inhaltsanteil content(d) („Ausprägung“) aufzuspalten und entsprechende Relationen sind für alle Fachobjekte zu definieren. Die Ausprägung von d wird mit v verknüpft, notiert als ver(d) ϵV, womit die Versionierung der Ausprägung realisiert wird.

Damit ist es konzeptuell möglich, die Veränderung eines mit logkey(d) identifizierten Objektes, über seine Versionen hinweg nachvollziehen und so ein Life-Cycle-Management für Objekte und damit ein versionierendes Repository realisieren zu können. Ein logischer Schlüssel logkey(d) wird in Ergänzung zu „klassischen“ Fachschlüsseln eingeführt, da diese im betrachteten Anwendungsfeld als teilweise veränderlich und damit nicht verlässlich sind.

Dieses Konzept schließt eine Mehrdeutigkeit von Ausprägungen unterschiedlicher Eigentümer zu einem Versionsvektor nicht aus, Inhalte einer Objektidentität können demnach mehrdeutig sein, d. h. formal für ein Fachobjekt fϵF eine Menge $D^f=\{d_1,\ldots,d_m\} \subseteq DS^f$ von Datensätzen existieren kann mit m>1, logkey(d_i)=logkey(d_j), ver(d_i)<>ver(d_j) und ver(d_i).e<>ver(d_j).e

1) Ist $z^{bis}_{z,i}$ undefiniert, entfällt die obere Grenze

Homepageveröffentlichung unbefristet genehmigt für VIA Consulting & Development GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt / © DVV Media Group GmbH



3: Exemplarischer Aufbau von ein- und ausgehenden Schnittstellen eines Systems zur versionierten Infrastrukturverwaltung. Der Datenlieferant kommuniziert die beschriebenen Daten und Statusänderungen. Intern hat das System eine Datenhaltung mit der bei diversen Prozessen lesend und schreibend kommuniziert wird. Datenempfänger erhalten auf Anfrage über eine weitere Schnittstelle die Daten vom System

(i,jϵ1,…,m, i<>j). Um eindeutige Ausprägungen einer Objektidentität zur Datenaufbereitung und Ausgabe bestimmen zu können, ist für alle Fachobjekte fϵF eine Selektionsfunktion sel $^{f}(D^{f})\epsilon D^{f}$ vorzusehen, die bei Mehrdeutigkeit der Ausprägungen diese auf eine gültige beschränkt und damit den Datensatz eindeutig macht.

Referenzen zwischen Datensätzen werden stets aufgrund logischer Schlüssel und damit unabhängig von konkreten Ausprägungen oder Gültigkeiten realisiert.

Konsistenzprüfung

Zwischen Datensätzen eines Fachobjekts sowie Fachobjektübergreifend können prozessbedingt Konsistenzprobleme und fehlende Passfähigkeit von Daten und Datensätzen auftreten, die von einem versionierenden System mit unterschiedlichen Dateneigentümern abgefangen werden können und sollten.

Setzt man voraus, dass Daten einer Bereitstellung in sich als konsistent gepflegt durch den Eigentümer gelten dürfen, kann die Passfähigkeit von Datensätze von Bereitstellungen aus der Feder unterschiedlicher Eigentümer aufgrund nicht synchronisierter Prozesse und Semantiken sowie abweichenden Wissens- und Bearbeitungsständen nicht gegeben sein.

Datensätze mit reduzierter Passfähigkeit können z. B. Topologie-Übergänge wie Betriebsstellengrenzen im eingeführten Beispiel sein, aber auch solche „Grenzpunkte nutzende“ Fachobjekte wie Fahrstraßen desselben Beispiels. In diesem kann zudem die dargestellte Fahrstraße prozessual nur dann durchgehend konsistent sein, wenn beide Eigentümer e1 und e2 zur selben Zeit dieselben Anpassungen an ihren jeweiligen Daten synchronisiert vornehmen und in Form einer Datenbereitstellung zum selben Zeitpunkt propagieren. Diese Situation ist im Beispiel in Bild 1 enthalten.

Für ein Fachobjekt fϵF, eine Menge M von Meldungen und seine Datensätze DS^{f} können deshalb Konsistenzprüfungen con: $DS^{f}xDS^{f}\rightarrow P(M)$ definierbar werden, die die Verletzung von Fachobjekt-spezifischen Semantiken offenbaren (mit Potenzmenge P(M) Meldungen).

Mit der Versionierung von Daten und Datensätzen werden zudem „hängende Referenzen“ relevant, die aufgrund des gewählten Konzepts von nicht-versionierten Objektidentitäten, versionierten Ausprägungen und Referenzen auf Objektidentitäten in Arbeitsprozessen ganz regulär entstehen können: Zu einem Versionsvektor V_z kann es Datensätze zu Versionen aus V_z geben, die Objektidentitäten referenzieren, zu denen es keine Ausprägungen zu Versionen aus V_z gibt, also „Daten ohne zum Zeitpunkt gültigen Inhalt“. Solche Problemsituationen können ebenfalls allgemeingültig definiert und mit Meldungen offenbart werden. Die Realisierung dieser Konsistenzprüfung erfolgt über die Definition von Prüfungsregeln ref: $DSxDS\rightarrow P(M)$ ohne Begrenzung auf Datensätze eines Fachobjekts.

Im begleitenden Beispiel in Bild 1 ist die gezeigte Fahrstraße ein Kandidat für die diversen Prüfungen. Ist die Blockteilung in Datenbereitstellungen von e_2 vor denen von e_1 enthalten, referenziert der Fahrstraßenbeginn auf bisher nicht mit

Homepageveröffentlichung unbefristet genehmigt für VIA Consulting & Development GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt / © DVV Media Group GmbH

Inhaltsdaten unterfütterte Objektidentitäten von „Signal Bk82“, da dieses erst mit einer entsprechenden Bereitstellung von e_1 übernommen und damit bekannt wird, die Referenz daher „hängt“. Stellt dementsprechend e_1 seine angepassten Daten vor e_2 bereit, basiert die Fahrstraße der e_2-Daten auf ebenfalls nicht aktuellen Daten, was über die Regel ref entsprechend offenbart werden sollte. Innerhalb der einzelnen Datenbereitstellungen kann über die Relation con zudem eine semantische Prüfung eines Fachobjekt-spezifischen Datenbestandes erfolgen, z. B. eine Prüfung, ob alle Signale einer Betriebsstelle eindeutige Namen haben etc.

Schnittstellen

Um eine Datenbereitstellung einspielen zu können, wird eine geeignete Eingangsschnittstelle benötigt, die verschiedene Minimalvoraussetzungen aufweisen muss. So müssen zu einer Bereitstellung dat=(e, s, z^{von}, z^{bis}, DS) die Metadaten e, s, z^{von},und z^{bis} bestimmbar sein, für die Fachobjekte der Datensätze müssen die beschriebenen Funktionen ver(d), owner(d), logkey(d), content(d) definiert sein. Mit diesen Grundvoraussetzungen ist dem versionierenden System eine Integration und Konsolidierung neuer Datenbestände bzw. Teilbestände unter Wahrung einer konsistenten Bereitstellung eines Gesamtdatenbestands möglich. Detektion einer Modifikation einzelner Datencharakteristiken ist mit weiteren Metadaten möglich. Sinnvoll sind zudem Schnittstellen, die den Statusübergang von Versionen abbilden. Diese können sowohl systemintern als auch von externen Quellen, wie zum Beispiel dem Datenlieferanten, nutzbar sein, ein Datenbestand also ggf. aufgrund seines Status eine „Reife-Historie“ von „vorläufig“ zu „freigegeben“ oder „zurückgezogen“ aufweisen. Diese Status können ggf. auch bei der Ausgabeschnittstelle Berücksichtigung finden („Projektierung auf vorläufigen Daten“).

Demgegenüber unterstützt eine Ausgangsschnittstelle verschiedene, von der Nutzung des versionierenden Systems abhängige Anfragen und Parametrisierungen. Wichtigster Parameter ist ein Zeitpunkt bzw. ein Zeitraum, für den die dann gültigen Daten gesamthaft bestimmt werden sollen. Weitere Angaben zu einer dedizierten Datenaufbereitung sind die Anfrage nach Eigentümern, bestimmten (Mindest-) Status, zu liefernder Fachobjekte und Aufbereitung der Ausgabedaten. Die Mächtigkeit der Anfrage richtet sich elementar nach Umfang und Granulari-

Homepageveröffentlichung unbefristet genehmigt für VIA Consulting & Development GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt / © DVV Media Group GmbH

Eurailpress DER EISENBAHN INGENIEUR ETR Rail business

Lassen Sie sich unverbindlich beraten!

DIGITALE SONDER | DRUCKE

Onlinemarketing mit Ihrem Fachartikel zur Nutzung in Ihren digitalen Kanälen

Werben Sie mit Ihrem maßgeschneiderten digitalen Sonderdruck!

Wir finden mit Ihnen die beste Ergänzung zu Ihrem Onlinemarketing-mix, sodass Sie Ihre digitale Reichweite optimal ausnutzen können.

Interesse? Ihre Ansprechpartnerin: **Martina Seemann**

lizenzen@dvvmedia.com | 040 237 14 139

DVV Media Group GmbH, Heidenkampsweg 73–79, 20097 Hamburg

DVV Media Group

Mehr Informationen unter **www.eurailpress.de/digitaler-sonderdruck**

tät der Realisierung und Mächtigkeit der verwalteten Datenbestände. Eine exemplarische Veranschaulichung wird in Bild 3 dargestellt.

Ausgabedaten werden aus allen Datensätzen abgeforderter Fachobjekte bestimmt, wobei die zur Versionierung von Datenausprägungen zuvor durchgeführte Aufteilung eines Datensatzes d in logische Schlüssel logkey(d) und Inhalte content(d) wieder rückgängig gemacht und auf einer vereinigten Datensatz reduziert werden. Dazu wird auch die bereits eingeführte Abbildung $sel^f(D^f) \epsilon D^f$ verwendet, die aus einer Menge von Ausprägungen eines Datenobjekts genau eine Ausprägung selektiert und einen Datensatz bildet.

Die Selektionsabbildung kann bei mehreren Datensätzen beispielsweise Daten eines bestimmten Eigentümers bevorzugen („Vorrangregel"), erste oder letzte Versionen eines Datensatzes bestimmen oder auch nach Versionen oder Eigentümern eines Datensatzes priorisieren.

Bei der Anfrage eines Zeitraums sind ggf. verschiedene Versionen eines Datensatzes – unter Beachtung der Selektionsabbildung – Teil der Ausgabedaten. Werden Ausgabedaten nach Zeiträumen angefordert, betrifft dies mehrere Versionsvektoren und denen zugehörige Daten können nach Eigentümern oder chronologisch geordnet aufbereitet werden. Ausgabedaten werden in Datenpaketen bereitgestellt.

Anwendungsfall PRISMA

Im Projekt PRISMA wird die Systemkomponente LeiDa-D zur versionierten Datenbereitstellung eines (Deutschland-) netzweiten Infrastruktur- und Basisdatenbestandes entwickelt. Diese Komponente stellt eine konkrete Realisierung der zuvor beschriebenen allgemeingültigen Ansätze und Konzepte dar. Einzelaspekte sind dabei beispielhaft:

- LeiDa-D führt die Datenbestände der Infrastruktur- und Basisdaten aller aus den Bestandssystemen LeiDa-S stammenden Betriebsstellen zu einem netzweiten Datenbestand zusammen. Die Daten stammen aus autonom arbeitenden Regionen, die als Eigentümer ihrer Datenbestände gelten.
- Einzelne Regionen stellen „ihre" Daten zur Verfügung, modellieren als Vorschau- bzw. Vormeldebereiche auch Daten benachbarter Regionen, die nicht zu übernehmen und deshalb zu filtern sind (Selektion auf Eigentümer).
- Randbereiche zwischen Regionen müssen auf Passfähigkeit und Konsistenz zur Sicht der Nachbarregion geprüft werden.
- Grundlegende Organisationsform ist die Betriebsstelle, in der Daten und Fachobjekte zusammengefasst werden. Betriebsstelle besitzen Eigentümer, die sich auf die enthaltenen Daten überträgt.

Die Daten einer Region werden als Datei eines proprietären Textdateiformats bereitgestellt. Nach Datenanalyse der aus Altsystemen stammenden Daten zeigt sich, dass für alle Basisdaten-Fachobjekte logische Schlüssel bestimmbar sind, die oft (aber nicht immer) dem herkömmlichen Verständnis fester Fachschlüssel entsprechen. Infrastrukturdaten können über regionseigene Schlüssel über ihre Lebenszeit hinweg identifiziert werden und sind daher zusammen mit Regionskennungen als logische Schlüssel nutzbar.

Basisdaten werden grundsätzlich in allen Regionen redundant und gleich verwendet und durch Dateien ebenfalls nach Eigentümer redundant bereitgestellt. Die Selektionsabbildung $sel^f(D^f)$ definiert deshalb eine „Vorrangregel", die Daten einer bevorzugten Region falls vorhanden auswählt. Als Ausgabeformat wird wiederum ein proprietäres Datenformat verwendet, welches den beschriebenen Eigenschaften der Ausgabeschnittstelle entspricht.

Anwendungsfall LaaS (LUKS as a Service)

LaaS ist eine serverbasierte Applikation, die vor allem aus LUKS (Leistungsuntersuchung Knoten/Strecken) bewährte Funktionalitäten wie Fahrzeit- und Belegungsrechnung sowie Konflikterkennung als Servicedienste zur kontextfreien Nutzung zur Verfügung stellt. LaaS stellt zudem eine Plattform dar, mit der die beschriebenen und vorgestellten Funktionalitäten eines Daten-versionierenden Systems ebenfalls als Service bereitstehen.

Die Kernfunktionalitäten lassen sich mithilfe offener API per HTTP-Kommunikation nutzen:

- Schnittstellen zur Anlage und Verwaltung versionsgeführter Repositorien;
- Konfiguration und Laufzeitverwaltung von Fachobjekten inkl. Schlüssel- und Inhaltsextraktion je Repository;
- Konfiguration und Parametrisierung von Konsistenz- und Prüfroutinen auf Fachobjekten;
- Einspielen mikroskopischer EBW (Eisenbahnbetriebswissenschaft)-Daten (derzeit XML-ISS (RailML/INT));
- Datenabfrage auf versionierte Daten und ableitbare Gesamtdatenbestände.

LaaS realisiert und evaluiert das eingeführte generische Konzept der Datenversionierung.

Ausblick und Bewertung

Das umrissene Konzept beschreibt Ansätze, Infrastrukturdaten wie Topologie, Sicherungstechnik und abgeleitete Nutzungsmöglichkeiten sowie zeitliche Ausprägungen und Basisdaten als zentrale Grundlage verschiedener Systeme des Bahnbetriebs (z. B. zur Disposition, Zugmeldung, Fahr- und Baubetriebsplanung, Analyse und Statistikerhebung) versioniert verwalten zu können.

Ausgehend von Erfahrungen und Erkenntnissen konkreter Projektarbeit sind damit generische Konzepte als allgemeingültige und generische Grundlage für Daten-versionierende Systeme für den dedizierten Anwendungsfall des Bahnbetriebs entwickelt und konkretisiert worden.

Mit den Erfahrungen aus dem diese Arbeiten initiierenden Projekt PRISMA werden die Ansätze in weitere Projekte übertragen und evaluiert. Im Projekt PRISMA werden sich die Umsetzungen bald im operativen Betrieb bewähren und beweisen, andere Projekte werden die Ansätze ggf. nachziehen oder schärfen helfen und sollen auch in operativen Systemen umgesetzt und gefestigt werden. ●

Summary

Versioning of infrastructure data and availabilities

Infrastructure data such as topology, interlocking information and derived usage options as well as temporal characteristics are the central basis of various systems of railway operations, e.g. for dispatching, train monitoring, timetable and construction planning, analysis and statistics. This article describes the concept and modeling of a versioned infrastructure management system.

Homepageveröffentlichung unbefristet genehmigt für VIA Consulting & Development GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt / © DVV Media Group GmbH