

XML-Schemata zum Datenaustausch im Planungsprozess von elektronischen Stellwerken

Die Verarbeitung von Planungsdaten auf Papier ist mit einem hohen manuellen Aufwand verbunden. Ein klar definiertes elektronisches, maschinenlesbares Austauschformat verspricht eine Effizienzsteigerung bei der Weiterverarbeitung und ermöglicht es, neue Wege bei der automatischen Validierung von Planungsdaten zu gehen.

→ Der Prozess der Planung und Projektierung von sicherungstechnischen Anlagen, insbesondere von elektronischen Stellwerken, ist komplex, da viele Organisationen und Personen beteiligt sind. Im Laufe des Planungsprozesses steigt der Detaillierungsgrad der Daten und die Daten werden zwischen den Beteiligten teilweise mehrfach ausgetauscht. In diesem Prozess ist es die Regel, dass Änderungen während der laufenden Planung notwendig werden, teilweise aufgrund erkannter Fehler, teilweise aufgrund geänderter Anforderungen an die LST-Anlagen. Die konkrete Ausprägung des Prozesses unterscheidet sich für jedes Land und ist durch nationale Richtlinien vorgegeben. Bei der Deutschen Bahn sind dies zum Beispiel die BAU-STE [1] und die Planungsrichtlinie 819 [2]. Den Abläufen ist aber in allen Ländern gemein, dass in einem definierten Prozessschritt die Planungsdaten vom Infrastrukturbetreiber an den Hersteller der LST-Anlagen übergeben werden.

1. MOTIVATION

Obwohl im Planungsprozess sowohl auf Seiten des Infrastrukturbetreibers als auch auf Seiten des Herstellers vielfach elektronische Werkzeuge eingesetzt werden, erfolgt die Prüfung sowie die Übergabe der Planungsdaten meist auf Papier. Damit ist ein hoher manueller Aufwand verbunden, zum einen bei der Prüfung, zum anderen bei der Übernahme der Daten auf der Herstellerseite in die eigenen Werkzeuge. Es ist also wünschenswert, den Datenaustausch in einem klar definierten elektronischen und maschinenlesbaren Format durchzuführen. Idealerweise kann dieses Format über den gesamten Planungsprozess verwendet werden.

Die Definition und Verwendung eines elektronischen Formats setzt wiederum eine stärkere Formalisierung des Prozesses voraus. So können Informationen, die heutzutage als Anmerkung oder Fußnote auf den Papierunterlagen stehen, nicht übernommen oder elektronisch ausgewertet werden. Zusätzlich müssen Daten aus Zeichnungen oder schematischen Lageplänen erfasst werden. Dies wird zum Beispiel bei der Betrachtung der in Deutschland üblichen Daten der Ausführungsplanung Planteil 1 (PT1) deutlich, die von der DB an die Stellwerkshersteller übergeben werden. Die relevanten Informationen sind nicht nur in den Tabellen wie Signaltabelle 1 und 2 oder der Fahrstraßentabelle enthalten. Ebenso wichtig ist die topologische Lage der Objekte in den Signallageplänen. Bei einer herstellerseitigen manuellen Übernahme der Daten in die eigenen Werkzeuge können diese Informationen durch den Menschen ausgewertet und erfasst werden. Bei einer elektronischen Übernahme der Daten müssen diese relevanten Informationen aber auch in das Übergabeformat aufgenommen werden. Das Datenformat muss flexibel genug sein, um Ausnahmen in der Planung oder Sonderkonstruktionen aufnehmen zu können, da dies in der Praxis oft vorkommt. Somit wird ein elektronisches Format benötigt, das flexibel ist, in dem aber dennoch keine falschen Daten übergeben werden können bzw. das entsprechend validiert wird. In diesem Artikel werden aktuelle Ansätze hierzu beschrieben und insbesondere eine Möglichkeit zur Validierung der Daten aufgezeigt.

2. XML-SCHEMATA IN DER EISENBAHNDOMÄNE

Als Sprache zur Definition von Schnittstellen hat sich in den letzten Jahren die vom World Wide Web Consortium (W3C) standardisier-



Dr. Dipl.-Phys. Carsten Gerke
Projektleiter Funkwerk Information Technologies GmbH

carsten.gerke@funkwerk-it.com



Dipl.-Inf. Joachim Bleidiessel
Softwareentwickler Funkwerk Information Technologies GmbH

joachim.bleidiessel@funkwerk-it.com



M.Sc. Michael Lodemann
wissenschaftl. Mitarbeiter
Institut für Informatik der Christian-Albrechts-Universität zu Kiel

milo@informatik.uni-kiel.de



Prof. Dr.-Ing. Norbert Luttenberger
Hochschullehrer
Institut für Informatik der Christian-Albrechts-Universität zu Kiel

nl@informatik.uni-kiel.de

te Extensible Markup Language (XML) [3] durchgesetzt. Eine Vielzahl an Werkzeugen unterstützt dieses Format. Die vom W3C herausgegebene Spezifikation definiert eine Metasprache, mit der strukturelle und inhaltliche Einschränkungen anwendungsspezifisch definiert werden können. Diese inhaltlichen Einschränkungen können z.B. durch die sogenannte Document Type Definition (DTD) [4] erfolgen. Ein erster Ansatz, auf XML-Basis mit Hilfe von DTD ein Datenformat zum Datenaustausch in der LST-Planung auf europäischer Ebene zu definieren, ist im Projekt Eurointerlocking [5,6] erfolgt. Hierbei wurde insbesondere das In-

terlocking Data File Format (IDAFF) definiert. Der Schwerpunkt lag dabei auf der Beschreibung aller für die Stellwerkslogik notwendigen Daten. Dieses Format hat allerdings keine große Verbreitung gefunden und wird im Wesentlichen von der ÖBB eingesetzt.

In den letzten Jahren wurde die DTD zunehmend durch die Definitionssprache XML-Schema (XSD) [7] verdrängt. Mit XML-Schema ist es ebenfalls möglich, die Struktur einer XML-Datei vorzugeben und inhaltliche Einschränkungen, wie z. B. Wertebereiche, zu definieren. XML-Schema ist dabei selbst wieder ein Satz von XML-Dokumenten und folgt der XML-Syntax. Viele Tools und Frameworks ermöglichen die syntaktische Validierung einer XML-Instanzdatei gegen ein XML-Schema. Nicht zuletzt wegen der im ersten Abschnitt genannten Flexibilität einer XML-Schnittstelle müssen aber weitergehende, semantische Prüfungen der XML-Daten erfolgen. Ein Ansatz hierzu wird unten vorgestellt.

In der Eisenbahndomäne gibt es mehrere Beispiele für die Verwendung von XML-Schemata zur Schnittstellendefinition. Eine relativ große Verbreitung hat railML [8] erlangt, das aktuell in der Version 2.1 vorliegt. Es handelt sich hierbei um ein Format unter der Creative Commons-Lizenz, das international sowohl bahn- als auch herstellerübergreifend verwendet wird. Allerdings wurde die Entwicklung von railML aus dem Bereich der Fahrplanplanung getrieben. Das railML-Schema gliedert sich daher in die Bereiche Allgemein, Infrastruktur, Fahrplan und Rollmaterial. Eine Fachdomäne Stellwerk ist zwar vorgeschlagen, existiert aber bisher nicht. Für den Datenaustausch zur LST-Planung genügt railML in seiner jetzigen Form deshalb nicht.

Bei der DB Netz AG in Deutschland wird im Projekt PlanPro, an dem Funkwerk als Hersteller der Stellwerkssimulation BEST [9,10] aktiv mitarbeitet, ein Datenformat auf Basis eines XML-Schemas für die LST-Planung definiert. Schwerpunkt sind dabei die Daten des PTI, die im derzeitigen Prozess auf Papier dem Stellwerkshersteller übergeben werden. Es werden nicht nur die für die Stellwerkslogik zwingend erforderlichen Daten aufgenommen, sondern auch weitergehende Informationen für die Errichtung des Stellwerks und der Außenanlagen, z. B. Verweise auf die zu verwendende Regelzeichnung. Wesentliche Aspekte des PlanPro-Schemas sind die klare Trennung von Topologie, Topographie und der darauf befindlichen LST-Objekte.

Auf europäischer Ebene beschäftigt sich das Projekt INESS [11] mit diversen Aspekten zur Definition eines neuen, einheitlichen Stellwerkskonzeptes für Europa. Eine Unterarbeitsgruppe des Projekts beschäftigt sich mit einem Datenmodell zur elektronischen Übergabe von LST-Planungsdaten. In einem ersten Schritt wird evaluiert, welches bereits bestehende XML-Format als Basis verwendet werden soll. In einem zweiten Schritt soll das

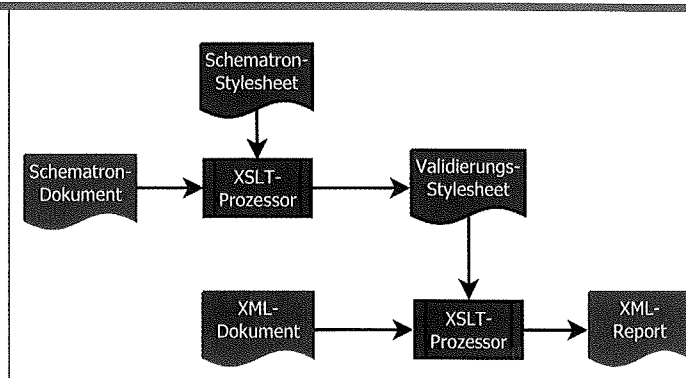


BILD 1:
Flussdiagramm
des technischen
Prozesses einer
Schematron-
Validierung
(Quelle aller Bilder:
Autoren)

gewählte XML-Format zum europäischen Datenmodell EUDRI (European Unified Description of Railway Infrastructures) erweitert werden. Informationen hierzu sind auf der Internetseite von INESS zu finden, auf der auch die öffentlich zugänglichen Berichte und Ergebnisse publiziert werden.

3. XML-SCHEMATA IM UNTERNEHMENSEINSATZ

Auch bei Funkwerk Information Technologies werden in vielen Bereichen Daten der Eisenbahninfrastruktur benötigt. Es wird derzeit ein Datenmodell entwickelt, mit dem (zunächst) Infrastrukturdaten unternehmensweit für die verschiedenen Produkte einheitlich gespeichert und ausgetauscht werden können. Das hiermit befasste Entwicklungsvorhaben AMBER (Advanced Model-Based Environment for Railways) geht über ein reines Datenaustauschformat weit hinaus. Es stellt eine komplette Umgebung zur Verfügung und soll die gesamte Produktkette über alle Projektphasen hinweg unterstützen.

Die Produktpalette von Funkwerk erstreckt sich dabei über große Teile der Eisenbahndomäne von Regional- und Rangierstellwerken über Dispositionssysteme bis hin zu Fahrplan-Planungssystemen. Dementsprechend muss das AMBER-Datenmodell sehr flexibel sein und je nach Anforderung unterschiedliche Abstraktionsgrade unterstützen. Diese verschiedenen Abstraktionsebenen dürfen wegen der deutlich unterschiedlichen Anforderungen nur lose aufeinander abgebildet werden. Für die Qualität eines Modells ist es entscheidend, Konsistenz- und Validierungsregeln aufstellen zu können. In AMBER wird dabei auf Modellebene die Object Constraint Language OCL der OMG [12] verwendet. Zur Persistierung und zum Datenaustausch kann aus dem AMBER-Modell ein XML-Schema generiert werden. Doch auch hier besteht die Notwendigkeit für eine semantische Überprüfung der Inhalte. Hierfür kann das im nächsten Abschnitt vorgestellte Schematron verwendet werden.

In dem Forschungsprojekt RTVE [13] wurde gemeinsam mit der Christian-Albrechts-Universität zu Kiel, gefördert durch das Land Schleswig-Holstein, auf Basis von Ontologien ein Konzept entwickelt, um Eisenbahn-

infrastrukturen verifizieren zu können. Mit Beginn des Projekts wählte man railML als zugrundeliegendes XML-Schema, da andere Projekte zur Findung eines einheitlichen Schemas noch nicht realisiert waren. In dem Forschungsprojekt wurde gezeigt, dass mit Hilfe von Ontologien die Verifikation der Infrastruktur erfolgen kann. Allerdings wird dabei das vorhandene XML-Schema in ein Ontologie-Modell transformiert. Die Synchronisation von Änderungen des XML-Schemas mit dem Ontologie-Modell stellte sich dabei als nicht trivial heraus.

Im Gegensatz dazu stellt die semantische Validierung mit der Schematron-Sprache eine Ergänzung von XML-Schema dar und soll deshalb im Folgenden vorgestellt werden.

4. SCHEMATRON

Schematron ist eine weitere Möglichkeit, XML-Inhalte auf Korrektheit zu überprüfen. Anders als XML Schema beschränkt Schematron die Validierung nicht allein auf die Struktur, die lokalen Abhängigkeiten und die Datentypen des Dokuments, sondern erlaubt weitergehende semantische Prüfungen anhand von Regeln. Schematron bietet sich deshalb als Ergänzung zur XML-Schema Sprache an. Dies soll anhand von Beispielen aus der Praxis demonstriert werden.

Schematron befindet sich seit 1999 in der Entwicklung und ist mittlerweile ein von der ISO verabschiedeter Standard [14]. Es existieren ausgereifte Werkzeuge, sowohl kommerziell als auch im Open-Source-Bereich, um Schematron einzusetzen.

XML-Schema prüft ein Instanzdokument nur syntaktisch, das heißt, es werden dessen Struktur und Datentypen überprüft. Schematron kann nicht nur die Struktur, sondern auch den Inhalt und die Bedeutung überprüfen. Es führt eine semantische Validierung des Dokuments durch.

Schematron wird in XML-Dokumenten häufig für die Überprüfungen von folgenden Sachverhalten eingesetzt:

1. Daten in einem XML-Dokument hängen voneinander ab. Zum Einen kann dies die Kardinalität betreffen. D. h. das Vorkommen eines Element- oder Attributinhalts bedingt die Anzahl der Vorkommen eines anderen Elements oder Attributs. Die- »

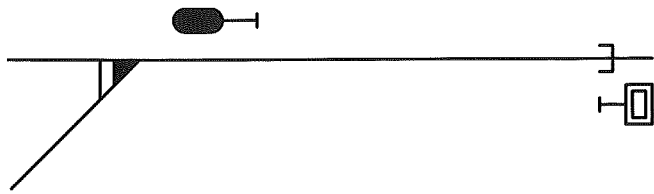


BILD 2:
Beispieltopologie
für die Validierung

se Abhängigkeit wird auf der Ebene des Instanzdokuments formuliert. In XML-Schema können nur Einschränkungen auf Klassenebene validiert werden. Zum Anderen können die Inhalte von Elementen und Attributen direkt von den Inhalten anderer Elemente und Attribute abhängen. Auch diese Abhängigkeit kann nicht in XML-Schema formuliert werden.

2. Zur Überprüfung der Korrektheit von Datensätzen in einem XML-Dokument müssen Berechnungen ausgeführt werden. Mit dem Ergebnis der Berechnung lässt sich eine Aussage über Behauptungen treffen.

Im Gegensatz zu XML-Schema erfolgt die Validierung nicht anhand einer formalen Grammatik, sondern beruht auf dem Prinzip der musterbasierten Suche (engl. „pattern matching“). Die Regeln in Schematron geben immer auch ein Suchmuster an. Wird ein In-

stanzdokument gegen die Regeln validiert, so wird in dem Dokument nach den Vorkommen von Elementen gesucht, die diesem Suchmuster entsprechen. Erfüllen die gefundenen Elemente die Regeln, ist das Dokument valide. Als Ergebnis wird ein Report erstellt. Dieser Report gibt an, welche Regeln erfolgreich geprüft wurden und wo es zu Fehlern kam.

4.1. DIE SCHEMATRON-TECHNOLOGIE

Die Sprache zur Formulierung von Validierungsregeln besteht nur aus wenigen Ausdrücken und ist deshalb einfach einzusetzen. Die Validierungsregeln werden in einem Schematron-Dokument formuliert. Die Angabe der Suchmuster und die Formulierung der zu prüfenden Bedingungen für eine Validierungsregel basieren auf den Technologien XSLT [15] und XPATH [16].

Ein Schematron-Dokument besteht aus

mehreren Regelblöcken, den „patterns“. Die Regelblöcke dienen der sinnvollen Gruppierung und Gliederung von ein oder mehreren Regeln, den „rules“. Jede Regel enthält Behauptungen, die asserts. Die Behauptungen enthalten letztlich die zu überprüfenden Validierungsregeln.

Ein XML-Dokument wird gegen ein Schematron-Dokument geprüft, indem die Validierungsregeln der Behauptungen im Kontext der Suchmuster geprüft werden.

Die technische Umsetzung dieses Validierungsprozesses ist in Bild 1 beschrieben. Das Schematron-Dokument wird zusammen mit einem vorher definierten Schematron-Stylesheet von einem XSLT-Prozessor zu einem Validierungs-Stylesheet transformiert. Dieses resultierende Validierungs-Stylesheet wird für die folgende Validierung genutzt. Dafür wird das zu prüfende XML-Dokument zusammen mit dem generierten Validierungs-Stylesheet durch den XSLT-Prozessor in einen XML-Report transformiert.

Dieser XML-Report ist in einer maschinenlesbaren Form formuliert. Der Report kann also insbesondere auch wieder weiterverarbeitet werden. So ist z. B. eine automatisierte Validierung möglich: Dokumente werden vom Werkzeug als nicht valide erkannt und das XML-Dokument kann z. B. abgelehnt werden. Auch ein Editor für das XML-Dokument kann so angepasst werden, dass die im Report erkannten Fehler im Dokument angezeigt werden.

Im folgenden Abschnitt soll der Einsatz von Schematron an einem Beispiel verdeutlicht werden.

4.2. SCHEMATRON IN DER PRAXIS

In der Praxis gibt es Anforderungen an Datenaustauschformate, die von der XML-Schemasprache nicht abgebildet werden können. Im Folgenden sollen beispielhaft drei Überprüfungen dargestellt werden, die einen Ausschnitt einer Eisenbahninfrastruktur verwenden. Diese Beispieltopologie und die XML-Repräsentation im AMBER-Datenmodell sind in den Bildern 2 und 3 dargestellt.

Das Beispiel beschreibt ein Stumpfgleis mit einer Länge von 1000 m, das am stumpfen Ende durch einen Bremsprellbock mit einem Sperrsignal begrenzt wird. Ein Ks-Signal mit den Signalbegriffen Hp 0 und Ks 1 ist am anderen Ende als Ausfahrtsignal vorhanden. Die an das Stumpfgleis angrenzende Weiche ist zur Verdeutlichung in der Grafik vorhanden, wird aber in diesem Beispiel zur Wahrung der Übersichtlichkeit im Datenmodell nicht mit angegeben. Die Repräsentation im XML-Dokument wird im folgenden Absatz erklärt. Im AMBER-Datenmodell wird eine Gleiskante durch ein Element „edge“ repräsentiert, das einen Anfang („tail“) ein Ende („head“) und eine Länge („length“) besitzt. Die Objekte der Infrastruktur, wie z. B. Signal und Bremsprellbock, werden durch „locations“ auf einer

BILD 3: XML-Dokument mit der Beispieltopologie im AMBER-Format

```

1 <acore:Infrastructure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xmlns:acore="http://www.funkwerk-it.com/amber/core" xmlns:amber="http://www.funkwerk-it.com/amber">
3   <edges id="edge1" head="node1" tail="node2" length="1000000">
4     <locations xsi:type="amber:Signal" type="Main_Signal" name="A"
5       distanceFromTail="100000" direction="TAIL_TO_HEAD" id="signal1">
6       <aspects id="aspect1" type="Hp_0" />
7       <aspects id="aspect2" type="Ks_1" />
8     </locations>
9     <locations xsi:type="amber:Signal" type="Main_Signal"
10      distanceFromTail="1100000" direction="HEAD_TO_TAIL" id="signal2">
11      <aspects id="aspect3" type="Sh_2" />
12    </locations>
13    <locations xsi:type="amber:BufferStop" type="Friction_Buffer_Stop"
14      distanceFromTail="997000" direction="HEAD_TO_TAIL" id="bufferstop1" />
15  </edges>
16 </acore:Infrastructure>
    
```

BILD 4: Schematron-Dokument mit den Validierungsregeln

```

1 ...
2 <iso:pattern>
3   <iso:title>Validierung der Positionen auf den Gleiskanten</iso:title>
4   <iso:rule context="Locations">
5     <iso:let name="edgeLength" value="number(parent::node()/@Length)"/>
6     <iso:assert test="$edgeLength >=@distanceFromTail">FEHLER: Position von '{iso:value-of select="@id"/}'
7       liegt außerhalb der Grenzen der Gleiskante.
8     Länge Gleiskante: <iso:value-of select="parent::node()/@Length div 1000"/>,
9       Position: <iso:value-of select="@distanceFromTail div 1000"/>
10    </iso:assert>
11  </iso:rule>
12 </iso:pattern>
13
14 <iso:pattern>
15   <iso:title>Validierung von Hauptsignalen</iso:title>
16   <iso:rule context="Locations[@xsi:type='amber:Signal'][@type='Main_Signal']">
17     <iso:assert test="aspects[@type='Hp_0']">FEHLER: Das Signal <iso:value-of select="@id"/>
18       ist als Hauptsignal modelliert, kann aber nicht Hp 0 signalisieren.</iso:assert>
19   </iso:rule>
20 </iso:pattern>
21
22 <iso:pattern>
23   <iso:title>Validierung der Positionen von Bremsprellböcken</iso:title>
24   <iso:let name="frictionOverlap" value="4000"/>
25   <iso:rule context="Locations[@xsi:type='amber:BufferStop'][@type='Friction_Buffer_Stop']">
26     <iso:let name="edgeLength" value="number(parent::node()/@Length)"/>
27     <iso:assert test="(@direction='HEAD_TO_TAIL' and ($edgeLength >=@distanceFromTail + $frictionOverlap))
28       or (@direction='TAIL_TO_HEAD' and (@distanceFromTail - $frictionOverlap >= 0))">
29       Der Bremsprellbock ist zu nah am Gleisende positioniert.</iso:assert>
30   </iso:rule>
31 </iso:pattern>
32 ...
    
```

Gleiskante verortet, indem ihr Abstand zum Anfang der dazugehörigen Gleiskante angegeben wird („distanceFromTail“). Die Länge der Kante und der Abstand eines Objekts zum Anfang sind als natürliche Zahl in Millimetern angegeben.

Intuitiv ist es klar, dass der Abstand eines Objekts zum Kantenanfang nicht länger als die Kante selbst sein darf. Überraschenderweise lässt sich dieser einfache Sachverhalt in der XML-Schema-Sprache nicht ausdrücken. In Bild 4 ist ein Schematron-Dokument zur Überprüfung der Topologie abgebildet.

In Schematron wird diese Prüffregel in den Zeilen 2 bis 12 im ersten Musterblock formuliert, der folgende Behauptung enthält: Für jede „location“ gilt, dass der Abstand zum Anfang der Kante immer kleiner oder gleich der Länge der Kante selbst ist.

Wird das XML-Instanzdokument gegen das Schematron Dokument validiert, so entsteht ein Report. Dieser Report ist für das Beispiel in Bild 5 dargestellt.

Die Zeilen 3 bis 13 sind das Ergebnis der Überprüfung der ersten Regel. In den Zeilen 6-12 ist zu erkennen, dass die Behauptung für das Signal mit der ID „s2“ fehlgeschlagen ist. Es liegt fälschlicherweise 100 m außerhalb der Gleiskante. Dies ist eine klassische Abhängigkeit zwischen den Dateninhalten von zwei Attributen.

Eine weitere Abhängigkeit besteht zwischen dem Typ eines Signals und dessen Signalbegriffen. Handelt es sich um ein Hauptsignal, so muss es zumindest den Signalbegriff Hp 0 signalisieren können. Der Typ eines Signals wird in AMBER mit einem Attribut angegeben. Die Signalbegriffe sind Unterelemente eines Signals, aspects bezeichnet. Diese Behauptung ist in Bild 4 in den Zeilen 14 bis 20 formuliert: Im Kontext einer location der Art Signal und des Typs Hauptsignal („Main_Signal“) gilt, dass ein Signalbegriff („aspect“) Hp 0 vorhanden sein muss. Im Beispiel wurde nun fälschlicherweise das Sperrsignal mit der ID „s2“ als Hauptsignal modelliert. Dies ist möglich, da das Datenmodell jede Kombination von Signalbegriffen erlaubt, um flexibel jede Art von Signal abbilden zu können. Dieser Fehler wird im Report in Bild 5 in den Zeilen 14 bis 21 offenbart.

Mit Schematron sind auch komplexe Überprüfungen möglich. Diese können eventuell von Projekt zu Projekt verschieden sein. Eine mögliche Anforderung in einem Projekt ist, dass ein Bremsprellbock mindestens 4 m vom Gleisende entfernt sein muss. Andernfalls kann er systembedingt seine Funktion nicht erfüllen. Zur Überprüfung dieser Behauptung muss eine Berechnung durchgeführt werden. Die Formulierung dieser Regel ist in Bild 4 in den Zeilen 22 bis 31 dargestellt: Zuerst wird in Zeile 24 dieser Mindestabstand als „frictionOverlap“ auf 4 m definiert. Es wird dann in Zeile 25 festgelegt, dass diese Regel nur im Kontext von Bremsprellböcken

```

1 <svrl:schematron-output>
2 ...
3 <svrl:active-pattern document="" name="Validierung der Positionen auf den Gleiskanten"/>
4 <svrl:fire-rule context="Locations"/>
5 <svrl:fire-rule context="Locations"/>
6 <svrl:failed-assert test="$edgeLength &gt;= @distanceFromTail"
7   location="/*:Infrastructure[1]/edges[1]/Locations[2]"/>
8   <svrl:text>FEHLER: Position von 's2'
9     liegt außerhalb der Grenzen der Gleiskante.
10    Länge Gleiskante: 1000,
11    Position: 1100</svrl:text>
12 </svrl:failed-assert>
13 <svrl:fire-rule context="Locations"/>
14 <svrl:active-pattern document="" name="Validierung von Hauptsignalen"/>
15 <svrl:fire-rule context="Locations[@xsi:type='amber:Signal'][@type='Main_Signal']"/>
16 <svrl:fire-rule context="Locations[@xsi:type='amber:Signal'][@type='Main_Signal']"/>
17 <svrl:failed-assert test="aspects[@type='Hp_0']"
18   location="/*:Infrastructure[1]/edges[1]/Locations[2]"/>
19   <svrl:text>FEHLER: Das Signal s2
20     ist als Hauptsignal modelliert, kann aber nicht Hp 0 signalisieren.</svrl:text>
21 </svrl:failed-assert>
22 <svrl:active-pattern document="" name="Validierung der Positionen von Bremsprellböcken"/>
23 <svrl:fire-rule context="Locations[@xsi:type='amber:BufferStop'][@type='Friction_Buffer_Stop']"/>
24 <svrl:failed-assert test="(@direction='HEAD_TO_TAIL' and ($edgeLength &gt;= @distanceFromTail + $frictionOverlap))
25   or (@direction='TAIL_TO_HEAD' and (@distanceFromTail - $frictionOverlap &gt;= 0))"
26   location="/*:Infrastructure[1]/edges[1]/Locations[3]"/>
27   <svrl:text>
28     Der Bremsprellbock ist zu nah am Gleisende positioniert.</svrl:text>
29 </svrl:failed-assert>
30 </svrl:schematron-output>

```

BILD 5: XML-Report als maschinenlesbares Ergebnis der Validierung

(„Friction_Buffer_Stop“) gilt. Die endgültige Behauptung ist in den Zeilen 27 bis 29 in folgender Weise definiert: Wirkt der Bremsprellbock in Richtung Gleiskantenanfang, so muss die Länge der Gleiskante beginnend beim Prellbock in Richtung Gleiskantenende noch mindestens 4 m lang sein. Wirkt der Bremsprellbock in Richtung Gleiskantenanfang, so muss die Länge der Gleiskante beginnend beim Prellbock in Richtung Gleiskantenanfang noch mindestens 4 m lang sein. Im Beispiel ist dieser Abstand mit 3 m zu klein bemessen. Im Report in Bild 5 wird deshalb in den Zeilen 24 bis 29 ein Fehler offenbart.

5. FAZIT

Es zeigt sich, dass XML im Zusammenspiel mit XML-Schema sehr gut für die Definition von Datenaustauschformaten im Bereich der sicherungstechnischen Planung geeignet ist. Es hat sich aber herausgestellt, dass XML-Schema allein nicht mächtig genug ist, um alle Abhängigkeiten innerhalb eines Modells auszudrücken. Semantische Abhängigkeiten, die sich durch die Betrachtung des Dateninhalts offenbaren, können mit XML-Schema nicht hinreichend validiert werden. Die Schematron-Technologie stellt hierfür eine passende Ergänzung innerhalb der XML-Familie dar, für die bewährte Werkzeuge vorhanden sind. Schematron erlaubt, Berechnungen über den Inhalt eines Dokuments durchzuführen. Zusammen mit XML-Schema realisiert es ein zweistufiges Vorgehen bei der Validierung von XML-Instanzdokumenten: XML-Schema wird für die Struktur und Schematron für den Inhalt eingesetzt. ←

Literatur

- [1] Verwaltungsvorschrift für die Bauaufsicht über Signal-, Telekommunikations- und Elektrotechnische Anlagen (VV BAU-STE 4.51)
- [2] Deutsche Bahn AG: LST-Anlagen planen – Richtlinie 819
- [3] W3C-Recommendation: Extensible Markup Language XML, <http://www.w3.org/TR/xml/> – Stand November 2008
- [4] DOCTYPE-Spezifikation in der XML-Spezifikation der W3C, <http://www.w3.org/TR/REC-xml/#dt-doctype> – Stand November 2008

- [5] Euro-Interlocking Projekt der UIC: <http://www.uic.org/spip.php?article391>
- [6] Johann Berger,Stephan Tschorn: Euro-Interlocking Pilot Projekt Data Preparation bei den Österreichischen Bundesbahnen, S + D 11/2005
- [7] W3C-Recommendations: XML-Schema Part 0 bis 2, <http://www.w3.org/TR/xmlschema-0/>, <http://www.w3.org/TR/xmlschema-1/>, <http://www.w3.org/TR/xmlschema-2/>, Stand Oktober 2004
- [8] RailML-Initiative: <http://www.railml.org/>
- [9] Olaf Seemann / Jörg Demitz: Betriebs- und Stellwerksimulation in der ESTW-Planung – Pilotprojekt Dresden-Neustadt, S + D 04/2008
- [10] Dipl.-Ing. Markus Grimm, Dipl.-Ing. Olaf Seemann, Jörg Demitz: Betriebliche Prüfung des ESTW Dresden-Neustadt mittels Simulation, ETR 10/2008
- [11] INtegrated European Signalling System INESS: <http://www.iness.eu/>
- [12] Object Constraint Language OCL der Object Modelling Group OMMG: <http://www.omg.org/spec/OCL/>
- [13] Lodemann/Gerke/Luttenberger: Beschreibung von Eisenbahninfrastrukturen mit railML und ihre Verifikation, S + D 4/2010
- [14] ISO/IEC 19757-3:2006. Information technology -- Document Schema Definition Languages (DSDL) -- Part 3: Rule-based validation -- Schematron [http://standards.iso.org/ittf/PubliclyAvailableStandards/co40833_ISO_IEC_19757-3_2006\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/co40833_ISO_IEC_19757-3_2006(E).zip)
- [15] W3C-Recommendations: XSL Transformations (XSLT) Version 2.0, <http://www.w3.org/TR/xslt20/>, Stand Januar 2007
- [16] W3C-Recommendations: XML Path Language (XPath) 2.0 (Second Edition), <http://www.w3.org/TR/xpath20/>, Stand Januar

SUMMARY

XML schemas for exchanging data in the process of planning electronic interlocking installations

Verification and transfer of planning data in the process of planning and engineering of safety-related installations is mainly done on paper, which involves a considerable amount of manual work and time. The industrial standard XML, together with XML Schema, enables definition of an electronic interchange format for planning data. Current developments in the railway sector are described in the article. However, content-related conditions cannot be adequately verified using XML Schema. Schematron technology, which enables content-related verification of electronic planning data, is presented for this task. Examples are given to illustrate this process.