

## Zukünftiger Aufbau der Verkehrsleittechnik gemäß den Technischen Lieferbedingungen für Streckenstationen (TLS)

FA 3.367

Forschungsstelle: Heusch/Boesefeldt GmbH, Aachen

Bearbeiter: Pögel, E. / Kaltwasser, J.

Auftraggeber: Bundesministerium für Verkehr, Bau und Stadtentwicklung, Bonn

Abschluss: August 2006

### 1. Aufgabenstellung

Im Rahmen dieses Projekts sollte die Definition der den Technischen Lieferbedingungen für Streckenstationen (TLS) zugrunde liegenden Daten in einer möglichst implementierungsunabhängigen Datenbeschreibungssprache unternommen werden.

Bis zur aktuellen TLS-Version (BMVBW, 2002) sind die Datenstrukturen der TLS in den textuellen Beschreibungen der TLS sowie den Bit-/Bytedefinitionen der Austauschtelegramme enthalten. Hierbei werden kommunikations- und anwendungsbezogene Festlegungen oft miteinander vermischt, was die TLS zu einem nur umständlich handhabbaren, vor allem aber schwer zugänglichen Regelwerk macht.

Neuere Verfahren der Datenmodellierung und positive Erfahrungen der letzten Jahre im Bereich der Modellierung von intelligenten Transportsystemen (ITS) legen es nahe, dieses Verfahren für zukünftige TLS-Versionen zu überdenken, um die TLS besser auf zukünftige Anforderungen auszurichten.

Unter Berücksichtigung des Stands der Technik in der Datenmodellierung und den Entwicklungen der Standardisierung in den Nachbarbereichen der TLS – namentlich zu nennen sind hier MARZ, DATEX, OKSTRA und OCIT – wurde eine geeignete Verfahrensweise ausgewählt und spezifiziert. Diese wurde dann zur Modellierung der existierenden TLS-Datenstrukturen verwendet.

Darauf aufbauend wurde die Umsetzung der DE-Blöcke der TLS in ein plattformunabhängiges Datenmodell (*platform independent model*, "PIM") einschließlich der Informationen für die Rückwärtsabbildung in die bestehenden TLS-DE-Blöcke, also das "PSM" (*platform specific model*) für die TLS 2002 Plattform vorgenommen.

Das Datenmodell selbst ist im Rahmen der Forschungsarbeiten werkzeuggestützt erstellt worden.

### 2. Lösungsansätze

Vergleicht man die aktuelle TLS mit verwandten Standards, wie z. B. den europäischen DATEX Vornormen für den Datenaustausch zwischen Verkehrszentralen oder der Entwicklung des OCIT Standards für städtische Systeme der Verkehrserfassung und Beeinflussung, so fällt auf, dass diese Systeme genau einen Schritt wie den hier beschriebenen bereits getan haben, beziehungsweise gerade dabei sind, eben dies zu tun.

Bei der zurzeit sich in Bearbeitung befindenden, zweiten DATEX Generation ist das textuell/tabellarisch beschriebene DATEX-Datenmodell durch ein grafisch definiertes Modell in der Modellierungssprache UML (*Unified Modelling Language*) ersetzt worden. Die EDIFACT Telegramme wurden dabei durch eine XML-Serialisierung der Objekte dieses UML-Modells ersetzt, welche formal über ein XML-Schema beschrieben ist. Dieses Schema wird mit Hilfe einer in Software implementierten Abbildung automatisch aus dem UML-Modell gewonnen.

Das hier gewählte Vorgehen vermeidet weitgehend Interoperabilitätsprobleme durch Interpretationsspielräume bei der Umsetzung des Datenmodells. Es birgt den entscheidenden Vorteil, dass es in üblichen Softwareentwicklungsumgebungen bereits in den Standardbibliotheken implementiert ist und nicht erst von den Anwendungsentwicklern in der Verkehrstelematik kosten- und fehlerträchtig umgesetzt werden muss.

Eine solche Vorgehensweise sollte daher auch bei der Überarbeitung der zukünftigen TLS angestrebt werden. Dabei stellte sich zunächst die Frage, welche Methode, welches Metamodell und welche Notation für die Weiterentwicklung der TLS den größten Nutzen versprechen.

Neben der bereits anhand des Beispiels DATEX erwähnten UML, kamen noch folgende Verfahren zur Beschreibung der TLS-Daten in Betracht:

- ASN.1 (siehe ISO 2002) definiert einen internationalen Standard für die Definition der Transfersyntax in der Darstellungsschicht von Kommunikationssystemen entsprechend dem ISO/OSI Referenzmodell.
- Weiter wäre eine direkte Modellierung als XML-Schema Definition möglich. Wenn, wie oben bereits am Beispiel DATEX geschildert, UML-Modelle zur Spezifikation der Übertragungssyntax häufig in XML-Schema Definitionen überführt werden, kann man sich fragen, ob man sie nicht direkt

als XML-Schema definieren sollte. Moderne XML-Werkzeuge bieten hier heute auch grafische Notationen zur Darstellung der Modelle an.

- Weiterhin sind aus der Vergangenheit diverse dedizierte Daten- bzw. Systemmodellierungsmechanismen bekannt, die oft im Umfeld von Datenbanksystemen Verwendung fanden, wie z. B. das Entity-Relationship-Modell (ERM) oder die Nijssen Information Analysis Method (NIAM), welche z. B. im OKSTRA als Ausgangsnotation Verwendung findet.

Für die Auswahl der Modellierungstechnik wurden folgende Aspekte herangezogen:

- 1) Verwendung einer modernen, in der Praxis weit verbreiteten Methode mit dem Ziel einer hohen Akzeptanz bei Entwicklern sowie einer guten Werkzeugunterstützung
- 2) Die Verwendung einer formalen Entwurfsmethode um sowohl eine hohe Eindeutigkeit der Spezifikation zu erreichen als auch die Möglichkeit, abstrakte Spezifikationen automatisiert auf Konstrukte der Implementierungsplattform(en) umsetzen zu können.
- 3) Die Verwendung einer intuitiven, bevorzugt grafischen Repräsentation von Konzepten, und vor allem ihrer Beziehungen untereinander, sorgt für ein besseres Verständnis und eine allgemein niedrigere Einstiegsschwelle in die Spezifikation.
- 4) Obwohl kein Muss, würde eine auch in den "Nachbarstandards" der TLS verbreitete Methode/Notation die Definition der Bezüge untereinander (bzw. der Schnittstellen zueinander) vereinfachen.
- 5) Schließlich musste der Aspekt berücksichtigt werden, inwieweit eine zu bewertende Methode die Spezifikation der Rückabbildung auf die existierende TLS 2002 Telegrammplattform unterstützt (Rückwärtskompatibilität).

Die Zusammenfassung der Bewertungen der vorgestellten Methoden ergab eine deutliche Präferenz für die UML, welche vor allem die Kriterien Plattformunabhängigkeit, eingängige Notation und Erweiterbarkeit sehr gut erfüllt.

Die direkte Modellierung in XML – bevorzugt dann als XML-Schema – mag zurzeit noch weiter verbreitet sein, hat aber den entscheidenden Nachteil, dass die XML (im Zusammenspiel mit den XML-Schema Spezifikationen) damit selber zur Implementierungsplattform wird. Diese wird zwar zurzeit zugegebenermaßen von allen Herstellern von Betriebssystemen und Entwicklungsumgebungen unterstützt. Dies sollte aber nicht darüber hinwegtäuschen, dass XML/XML-Schemata keineswegs abstrakte Spezifikationsmethodiken sind, sondern ganz im Gegenteil sehr konkrete Syntaxnotationen!

ASN.1 und frühere Methodiken der Datenmodellierung wie ERM oder NIAM mögen in speziellen Bereichen ihre Vorteile haben, sind aber im hier betrachteten Kontext der Systementwicklung eher exotisch und haben viel zu wenig Verbreitung und Unterstützung, um die verfolgten Ziele zu erreichen.

Aus dem hier Gesagten blieb daher nur der Schluss übrig, die UML als Spezifikationsprache für die zukünftige TLS vorzuschlagen, wobei der automatischen Abbildung auf XML-Strukturen für den Austausch (XML-Schema) ein besonderes Augenmerk geschenkt werden sollte.

### 3. Anwendung der UML

Im Kontext der vorgeschlagenen TLS-Modellierung sollten die folgenden zwei Diagrammformen der UML zur Anwendung kommen:

#### 1. Klassendiagramme

Klassendiagramme dienen der Darstellung von Objektklas-

sen und ihrer Bezüge untereinander. Klassen können in der UML sowohl Attribute ("Zustand") als auch Methoden ("Verhalten") besitzen. Außerdem können über entsprechende Konzepte des UML-Metamodells eine textuelle Definition, Einschränkungen ("constraints") und Eigenschaftswerte ("tagged values") erfasst werden.

Folgende Klassenbeziehungen sollten im spezifizierenden Teil verwendet werden:

- Spezialisierungen werden für die Modellierung von Vererbungsbeziehungen genutzt. Um Implementierungsproblemen vorzubeugen wird das Überschreiben von Attributen ausdrücklich nicht vorgesehen, dass heißt eine spezialisierte Klasse kann den Zustandsraum der Vaterklasse nur durch neue Attribute erweitern, nicht jedoch ererbte Attribute ausblenden oder verändern.
- Kompositionen dienen der Beschreibung von Enthaltensein-Beziehungen, wobei Objekte der komponierten Klasse ohne die Ausgangselemente nicht lebensfähig sind.
- Aggregationen stellen ebenfalls eine Enthaltensein-Beziehung dar, in diesem Fall aber von Elementen die durchaus ein eigenständiges Leben führen.

#### 2. Sequenzdiagramme

Sequenzdiagramme stellen in der UML das Pendant zu den bereits in der aktuellen TLS verwendeten Pfeildiagrammen dar, welche Abläufe der Signalisierung (im objektorientierten Sinne also des Methodenaufrufs) zwischen Objekten im zeitlichen Ablauf skizzieren.

*Es sei an dieser Stelle darauf hingewiesen, dass lediglich ein kleiner Teil der möglichen Modellierungstechniken aus der UML derzeit verwendet wird. Insbesondere mittels Zustandsdiagrammen könnten wesentlich komplexere Zusammenhänge, z. B. Schaltungsabläufe in der FG 4, in den TLS beschrieben werden. Für die Modellierung der Daten der derzeitigen TLS, d. h. DE-Blöcke, war dies aber nicht erforderlich.*

### 4. Die Modellierung

Aufbauend auf diesen grundlegenden methodologischen Vorüberlegungen erfolgte die Ausformulierung des TLS-Datenmodells in UML.

Die Modellierung wurde werkzeuggestützt durchgeführt. Als Werkzeug für die Modellierung wurde das Tool Enterprise Architect (EA) des australischen Herstellers Sparx Systems Pty Ltd (<http://www.sparxsystems.com>) gewählt.

In einem ersten Modellierungsschritt wurde ein TLS-Systemmodell erstellt. Dieses Modell beschreibt exemplarisch den dreistufigen Aufbau eines TLS-Netzwerks bestehend aus VRZ, UZ und Streckenstation. Ferner ist eine Beschreibung der Objekte innerhalb einer Streckenstation vorgenommen worden. Das TLS-Systemmodell wurde allerdings nur insoweit ausformuliert, als es für die Modellierung von Abläufen zum Datenaustausch zwischen Objekten der TLS-Welt erforderlich ist.

Als zweiter Schritt wurde ein TLS-Profil für die Datenmodellierung erstellt. Dieses besteht aus Stereotypen zur Modellierung primitiver Datenkonzepte (Attributtypen). Es wird dazu verwendet, TLS-Daten in DE-Blockklassen zu formulieren. Zu nennen sind hier die Stereotype <<intervall>>, <<physikalisch>>, <<zeichenkette>> und <<zeitpunkt>>. Mit <<ablauf>> wurde ein weiterer Stereotyp festgelegt, der TLS-Metadaten für Methodenaufrufe enthält. Jeder dieser Stereotype enthält Eigenschaftswerte ("tagged values"), die zur Beschreibung von TLS-Daten in DE-Blockklassen benötigt werden.

Ursprünglich war vorgesehen, an dieser Stelle mit der Modellierung der TLS-DE-Blockklassen zu beginnen. Allerdings wurde schnell offensichtlich, dass dies nicht erfolgreich sein würde, ohne sich auch direkt mit der Rückwärtsabbildung der Klassen auf die bestehenden DE-Blöcke zu befassen. Aus diesem Grund wurde als nächster Schritt ein TLS-Profil für die Rückwärtsabbildung erstellt. Mit den Stereotypen dieses Profiles resp. den zugehörigen Eigenschaftswerten sollten die DE-Blockklassen und ihre Daten derart attribuiert werden, dass mittels eines Automaten, also einem "Stück Software", aus einem konkreten Objekt einer DE-Blockklasse wieder der zugehörige, aus der TLS bekannte DE-Block bitgenau konstruiert werden kann.

Diese beiden TLS-Profile bildeten den "Werkzeugkasten" für die eigentliche Modellierung der DE-Blockklassen zu den TLS-DE-Blöcken. I. W. wurde für jeden DE-Block ein eigenes Klassen-Package erstellt, diese Packages naheliegenderweise nach TLS-Funktionsgruppen wiederum in Packages gruppiert. Übergreifend verwendbare Festlegungen von Datentypen sowie abstrakten DE-Blockklassen wurden separat zusammengefasst. Dabei erfolgte begleitend zu den sehr umfangreichen Arbeiten der eigentlichen DE-Blockklassenmodellierung die nicht minder aufwändige Attributierung dieser Klassen im Hinblick auf die geforderte Rückwärtsabbildung der Klassenobjekte in TLS-DE-Blöcke.

Mit zwei weiteren Modellierungsschritten sollte gezeigt werden, wie Objekte eines TLS-Netzwerks mittels Objekten aus den DE-Blockklassen Daten austauschen können. Hierzu erfolgte zunächst die Modellierung von Austauschklassen auf der Grundlage des bereits genannten TLS-Systemmodells. So erhielt die Unterzentrale FG-spezifische "Softwareeinheiten", während in den Streckenstationen entsprechend der E/A-Kanal FG-spezifisch spezialisiert wurde. Die zugehörigen Klassen erhielten Methoden, über die Objekte der DE-Blockklassen zwischen UZ und Streckenstation ausgetauscht werden können. Hierzu wurden folgende Austauschprimitive DE-Block-spezifisch als Methodenaufrufe ausmodelliert:

1. "empfang"-Methoden: Dies sind die spontanen Meldungen der Ereignisklasse 1 nach TLS.
2. "frage"-Methoden: Hier fragt die Unterzentrale eine Information von einem DE ab (Ereignisklasse 2)
3. "setze"-Methoden: Diese fassen die Ereignisklassen 3 und 4 nach TLS zusammen.

Auf der Grundlage dieser Austauschklassen konnte die Erstellung von Sequenzdiagrammen erfolgen. Diese Diagramme beschreiben, wie die in Austauschklassen modellierten Methodenaufrufe verwendet werden, um DE-Blockklassenobjekte auszutauschen. Während die "empfang"-Methoden seitens der Streckenstation in Softwareeinheiten der UZ aufgerufen werden, werden "frage"- und "setze"-Methoden in der Streckenstation aufgerufen.

### 5. Schluss und Ausblick

Es wurde im FE-Projekt mittels des Ansatzes der modellgetriebenen Architektur das den TLS zugrunde liegende Datenmodell in UML erfasst. Darüber hinaus wurde der Austausch dieser Daten zwischen Straße und Unterzentrale ausmodelliert. Wichtiger Aspekt war dabei, dass auch die Rückabbildung auf die existierenden Telegrammdefinitionen der TLS 2002 gelingt.

Zusammen mit der hier vorausgesetzten Abbildung der Kommunikationsschichten 2 und 3 auf eine TCP/IP-basierte Netzinfrastruktur liegt damit eine eindeutige Spezifikationsgrundlage für den Mischbetrieb einer nach diesem Modell aufgebauten Verkehrstelematikarchitektur mit existierenden Anlagen vor.

Datenstrukturen und Schnittstellendefinitionen können aus einem solchen UML-Modell automatisch und eindeutig generiert werden, und über die plattformspezifische Abbildung auf die existierende Telegrammstruktur liegt eine eindeutige, rückwärtskompatible Abbildung der existierenden TLS-Abläufe vermöge Austausch von OSI7-Telegrammen vor.

Für zukünftige alternative Transfersyntaxen steht damit eine plattformunabhängige Beschreibung der Daten zur Verfügung, die mit Hilfe der UML-Erweiterungsmöglichkeiten ohne Paradigmenbruch zur Generierung entsprechender Schnittstellenspezifikationen genutzt werden kann.

Der objektorientierte Ansatz ist dabei nicht auf die Aspekte der Datenmodellierung beschränkt, sondern kann auch zur Modellierung anderer Aspekte, insbesondere der Ablauflogik Verwendung finden.

Insbesondere in diesem Zusammenhang ist darauf hinzuweisen, dass mit der vorliegenden Umsetzung der derzeitigen TLS in UML nicht annähernd das Potenzial dieser Modellierungstechnik ausgeschöpft werden musste. Gerade bei dem Thema "Abläufe" sind hier erhebliche Entwicklungsschritte zu erwarten. Als Beispiel sei hier nur die Möglichkeit angerissen, auch komplexe Schaltvorgänge im Bereich der derzeitigen FG4 abzubilden – und dabei auch noch zusätzlich transaktionsfest zu sein. Vergleichbare Perspektiven sind für Daten und die sie austauschenden verkehrstechnischen Objekte zu erkennen.

Im parallelen FE-Projekt 03.0385/2004/IRB "Zusammenwirken von betriebs- und verkehrstechnischen Tunnelausstattungs-elementen im Bereich von Straßentunneln" wird der Einsatz von TCP/IP als Kommunikationsinfrastruktur im Wirkungsbereich der TLS gefordert. Dabei soll sich der Einsatz eines TCP/IP-Netzwerkes nicht nur auf den Transport für derzeitige TLS-OSI7-Telegramme, wie es z. B. mit TLS-over-IP realisiert wird, beschränken. Vielmehr wird gefordert, dass bis hinauf zu ISO-OSI-Schicht 7 Standardtechnologien zum Einsatz kommen. Hierbei werden insbesondere derzeit gängige Architekturen unter den Stichworten CORBA oder SOA in den Vordergrund gestellt.

Aus dem gegenständlichen Forschungsprojekt wird diese Forderung und Tendenz nachhaltig unterstützt. Mittels der durchgeführten Umsetzung der bestehenden TLS in ein PIM sind die modelltechnischen Voraussetzungen für die Migration hin zu einer TCP/IP-basierten Infrastruktur im Wirkungsbereich der TLS geschaffen.

Als Ergebnis wird es nicht mehr erforderlich sein, dass in den TLS ein Protokoll unterhalb ISO-OSI-Schicht 7 spezifiziert ist. Insofern sollten auf der Grundlage der Forschungsergebnisse die "unteren" TLS-Schichten das Attribut "abgekündigt" erhalten, zukünftige Anlagen nach TLS nur noch über TCP/IP kommunizieren.

Es wird aber an dieser Stelle deutlich darauf hingewiesen, dass mit der hier vorliegenden Modellierung der bestehenden TLS in dieser Hinsicht durchaus noch nicht alle Voraussetzungen geschaffen sind. Insbesondere die Erstellung eines PSM – womit hier natürlich nicht die Rückabbildung auf die bestehende TLS gemeint ist – ist noch eine offene Aufgabe, die im Zusammenwirken von AK TLS und AK MARZ erfolgen muss.

Allerdings zeigen auch hier die Ergebnisse der beiden genannten Forschungsprojekte bereits wesentlich mögliche Richtungen auf. Es wird an dieser Stelle konkret vorgeschlagen, dass eine dienstebasierte Infrastruktur angestrebt wird. Hierzu bietet das jetzt in UML vorliegende TLS-DM die Voraussetzungen.

Eine naheliegende Umsetzung, die nachdrücklich unterstützt wird, ist die Realisierung einer SOAP-basierten Kommunikationsinfrastruktur. Die hierfür notwendige XML-Spezifikation der

Dienste, Abläufe und Datenstrukturen der TLS kann mittels der WSDL (Web Services Description Language) problemlos aus dem jetzt vorliegenden TLS-DM erfolgen. Die Kommunikation erfolgt über HTTP und TCP/IP. Auf dieser Grundlage sollte unverzüglich eine entsprechende Testimplementierung geschaffen werden.

Auch an dieser Stelle zeigt sich, dass der grundlegende Modellierungsansatz dieses FE-Projektes, im Hinblick auf die Schaffung eines Netzwerks miteinander kommunizierender Geräte mit verkehrstelematischen Aufgaben durch die Trennung von Inhalt und Kommunikation in der TLS einen wesentlichen Beitrag zu leisten, erreicht ist. Hiermit wird ein Weg aufgezeigt, wie es möglich ist, das verkehrstelematische Knowhow, das den Kern der TLS ausmacht, zukunftsfähig zu erhalten.

## Glossar

- ASCII – American Standard Code for Information Interchange; Zeichenkodierung in Rechnersystemen
- ASN.1 – Abstract Syntax Notation One; internationaler Standard für die Datenbeschreibung in Kommunikationssystemen
- AUSA – verwaltungseigenes, bundesweites Telefonnetz
- CEN – Europäisches Komitee für Normung
- CORBA – Common Object Request Broker Architecture; objektorientierte Middleware mit plattformübergreifenden Protokollen und Diensten
- DATEX – europäische Vornorm zum Datenaustausch zwischen Verkehrszentralen
- EDIFACT – Electronic Data Interchange For Administration, Commerce and Transport; branchenübergreifender internationaler Standard für das Format elektronischer Daten im Geschäftsverkehr
- ITS – Intelligente Transportsysteme
- LCC – Upper Case Camel, Namenskonvention, bei der alle Wortteile mit einem Großbuchstaben beginnen
- MARZ – Merkblatt für die Ausstattung von Verkehrsrechnerzentralen und Unterzentralen
- NIAM – Nijssen Information Analysis Method; Systementwurfstechnik, die ursprünglich aus der Linguistik stammt
- OCIT – Offene Schnittstellen für die Straßenverkehrstechnik, offener Industriestandard für Straßenverkehrstelematik im städtischen Raum
- OCL – Object Constraint Language; Standard der OMG
- OKSTRA – Objektkatalog für das Straßen- und Verkehrswesen

- OSI – Open Systems Interconnection; offenes Schichtenmodell für die Organisation von Kommunikationstechnik
- PIM – Platform independant model
- PSM – Platform specific Model
- SOA – Serviceorientierte Architektur
- SOAP – Simple Object Access Protocol; seit SOAP 1.2 ein Begriff in sich, da SOAP seither weder "einfach" ist noch nur dem "Zugriff" dient
- TLS – Technische Lieferbedingungen für Streckenstationen
- UCC – Upper Case Camel; Namenskonvention, bei der alle Wortteile außer dem ersten mit einem Großbuchstaben beginnen
- UML – Unified Modeling Language; standardisierte Beschreibungssprache für den Systementwurf
- WSDL – Web Services Description Language; plattform-, programmiersprachen- und protokollunabhängige Spezifikation
- XML – eXtensible Markup Language; Standard zur Kodierung maschinen- und menschenlesbarer Dokumente
- XMI – XML Metadata Interchange; OMG-Standard bzgl. Austauschformat zwischen Software-Entwicklungswerkzeugen

## Literaturverzeichnis

- BMVfW Bundesministerium für Verkehr, Bau- und Wohnungswesen: Technische Lieferbedingungen für Streckenstationen, Ausgabe 2002
- CEN Europäisches Komitee für Normung: DATEX Verkehrs- und Reisedatenvokabular (Version 3.1a), CEN ENV13106:2000, Mai 2000
- CEN Europäisches Komitee für Normung: DATEX Spezifikation für den Datenaustausch zwischen Verkehrs- und Reiseinformationszentralen (Version 1.2a), CEN ENV13777:2000, Mai 2000
- FOWLER, M.: UML Distilled: A Brief Guide to the Standard Object Modelling Language, 3. Auflage, Addison-Wesley Professional, 2003
- ISO Internationale Organisation für Standardisierung: Information Technology - Abstract Syntax Notation One (ASN.1), ISO/IEC 8824-1:2002 bis 8824-4:2002. 2002
- OESTERREICH, B.: Analyse und Design mit UML 2: Objektorientierte Softwareentwicklung, 7. Auflage, Oldenbourg Wissenschaftsverlag, 2004
- W3C World Wide Web Consortium: XML-Schema Part 1: Structure sowie XML-Schema Part 2: Datatypes, W3C Recommendation, Mai 2001