



**Machbarkeitsstudie
OKSTRA[®]-Modellierung mit UML und GML**

von

*interactive instruments Gesellschaft für Software-Entwicklung mbH
Trierer Straße 70-72
53115 Bonn*

im Auftrage der

*Bundesanstalt für Straßenwesen
Referat Z4
Brüderstraße 53
51427 Bergisch Gladbach*

Verfasser:

Dr. Jochen Hettwer

Bernd Weidner

Die verwendeten Firmen- und Produktnamen sind im allgemeinen eingetragene Warenzeichen der jeweiligen Eigentümer.

0 Inhaltsverzeichnis

0	Inhaltsverzeichnis.....	3
1	Einleitung.....	5
1.1	Ist-Stand der Modellgenerierung	5
1.2	Bewertung des Verfahrens aus heutiger Sicht	5
2	Technische Umsetzbarkeit.....	9
2.1	Abbildbarkeit des OKSTRA-EXPRESS-Modells nach UML.....	9
2.2	Umwandlung des Express-Codes in UML.....	11
2.3	GML und der MDA-Ansatz.....	12
2.4	Überarbeitung von Basismodellen.....	13
2.5	Zu modifizierende lexikalische und syntaktische Regeln	13
3	Auswirkungen der Umstellung auf Prozesse	15
3.1	Auswirkungen auf die Pflegestelle.....	15
3.2	Auswirkungen auf die Softwarehersteller	16
3.3	Auswirkungen auf die Anwender in Bund und Ländern.....	16
4	Auswirkungen der Umstellung auf Produkte.....	17
4.1	Schemata.....	17
4.2	Dokumentation.....	17
4.3	Wiki	18
4.4	Programm-Bibliotheken	18
4.5	Prüftool.....	18
4.6	Normausgabewerkzeug.....	18
4.7	OKSTRA-konforme Web Services.....	18
5	Migrationskonzept.....	19
5.1	Erhaltung der Kompatibilität mit alten Versionen.....	19
5.2	Akzeptanz bei den Betroffenen.....	19
5.3	Arbeitsschritte für die Migration	19
6	Resümee	20

1 Einleitung

1.1 Ist-Stand der Modellgenerierung

Der OKSTRA[®] wird von den Experten mit Unterstützung der Pflegestelle zunächst mit Hilfe von NIAM formuliert, wobei jedoch keine vollständige grafische Umsetzung erfolgt, um die Diagramme nicht platzen zu lassen. So werden Attribute nur in Ausnahmefällen dargestellt.

NIAM hat den Vorteil, die Schemata des OKSTRA in einer grafischen Notation abzubilden. Es kommt hierbei mit einer sehr kleinen Anzahl von grafischen Symbolen aus, ist aber dennoch mächtig genug, um auch komplexe Sachverhalte abzubilden. NIAM ist leicht erlernbar und eignet sich gut für die Modellierung in Arbeitsgruppen.

Die endgültige Referenzmodellierung wird durch eine händische Umsetzung der NIAM-Diagramme und weiteren Notizen zu den Attributen textbasiert in EXPRESS formuliert. Als Werkzeug wird hierzu ein normaler Texteditor benutzt. Aus dem Referenzmodell werden dann automatisch über Werkzeuge erzeugt

- XHTML-Seiten zur Dokumentation,
- daraus wiederum die Vorlagen zur Führung des OKSTRA[®]-Wiki,
- Quellcode für die XTRA-Software, die die Grundlage für das OKSTRA[®]-Prüftool bildet,
- das OKSTRA[®]-XML-Modell.

Da Attribute in den NIAM-Diagrammen nicht dargestellt werden, wird bei ihrer Behandlung in den Arbeitsgruppen entweder auf eine textliche (nicht formalisierte) Beschreibung oder auf die formalisierte Modellierung in EXPRESS zurückgegriffen. Dies ist insbesondere bei Überprüfungen zur Qualitätssicherung notwendig.

1.2 Bewertung des Verfahrens aus heutiger Sicht

Das Verfahren ist eingespielt und hat sich im Großen und Ganzen bewährt. Modellierungssprache EXPRESS und Data Encoding CTE orientieren sich an internationalen Standards (ISO 10303) und erfordern daher prinzipiell nicht die Bindung an bestimmte Produktlinien für die Verarbeitung von OKSTRA[®]-Daten.

In der Zeit seit der Entstehung der ersten Version des OKSTRA[®] haben sich jedoch neue technische Entwicklungen im Felde der Informationsmodellierung ergeben, so dass es geraten erscheint, die Möglichkeiten dieser Techniken im Hinblick auf den OKSTRA[®] zu untersuchen.

1.2.1 Stand der Technik: Modellierung und Datenkodierung

Auf dem Feld der Informationsmodellierung ist die Sprache *UML* (Unified Modeling Language), die von der Object Management Group OMG betreut wird, mittlerweile der beherrschende Standard geworden, während für die Repräsentation und die Übertragung strukturierter Daten die auf der Extensible Markup Language *XML* basierenden Standards dominierend geworden sind. Die Pflege dieser Standards obliegt dem World Wide Web Consortium W3C.

Neben der XML-Basisstandards, die die Kodierungsregeln, die Benennungsmechanismen und die Verknüpfungskonstruktionen für XML-basierte Dokumente spezifizieren, gibt es eine immens große Zahl von XML-basierten Anwendungsstandards. Für die Belange der Modellierung sind hierbei *XML Schema*, *RDF* (Resource Description Framework, ein Standard zur Metadatenmodellierung) und *OWL* (Web Ontology Language, ein RDF-Dialekt zur formalen Wissensrepräsentation) zu nennen. Wichtige Gründe für die große Verbreitung der XML-Standards sind erstens die Möglichkeit, relativ unkompliziert Formate für neue Anwendungsfelder konstruieren zu können, und zweitens die Möglichkeit, Formate für verschiedene Bereiche leicht miteinander kombinieren zu können. Auch universelle "Hilfssprachen", z.B. für Autorisierung (XACML), Interfacebeschreibung (WSDL), Mathematik (MathML) usw. (siehe <http://www.w3c.org> sowie <http://www.oasis-open.org/>)

sind mit XML implementiert. Mit Hilfe von XSLT steht außerdem eine Sprache zur Verfügung, mit deren Hilfe man aus Daten, die nach einer fachspezifischen Strukturbeschreibung in XML kodiert vorliegen, leicht weitere Daten für die verschiedensten Zwecke ableiten kann, z.B. für Internetpublikation in XHTML oder für den Gebrauch in Bürosoftware-Systemen in den ISO-Dokumentformaten ODF oder OOXML.

Die Dominanz von UML und XML hat dazu geführt, dass es eine Vielzahl von Software-Werkzeugen, Fachbüchern und Internet-Ressourcen gibt und dass die Schulung darin zum Standardrepertoire einer Informatikausbildung gehören.

In der überwiegenden Zahl der Fälle werden Informationsmodellierungen deshalb heutzutage unter Verwendung dieser Standards erstellt. Einige wenige Beispiele hierfür sind:

- 3A(AFIS-ALKIS-ATKIS), der Standard des deutschen Vermessungswesens für die Modellierung von Liegenschaften und Topografie, siehe <http://www.adv-online.de>,
- AIXM (Aeronautical Information Exchange Model), der Standard für Daten in der Luftfahrt, siehe <http://www.aixm.aero>
- CSML (Climate Science Modelling Language) der Standard für Atmosphären- und Ozeanographiedaten, siehe <http://csml.badc.rl.ac.uk/>
- CityGML, der Standard für dreidimensionale Stadtmodelle, siehe <http://www.citygml.org/>
- IFOPT (Identification of Fixed Objects in Public Transport), ein Standard zu Modellierung von Einrichtungen im öffentlichen Verkehr, siehe <http://www.ifopt.org>
- Dublin Core, ein Standard für Metadaten etwa zu Publikationen), siehe <http://dublincore.org/>
- die unter der Bezeichnung XÖV bekannten Standards für das deutsche eGovernment (siehe <http://www.osci.de>, <http://www.deutschland-online.de> und <http://www.kbst.de>)

Insgesamt kann man sagen, dass das bisher im OKSTRA® verwendete STEP/EXPRESS im Vergleich zu den genannten Technologien nicht sehr verbreitet ist. Es gibt eine Gemeinschaft von Entwicklern und Herstellern (<http://www.prostep.org/>), die sich weiterhin aktiv mit der Modellierung von Produktinformation beschäftigt. Auch vor diesem Bereich, aus dessen Anforderungen heraus STEP/EXPRESS entstanden ist, haben die Technologien UML und XML nicht haltgemacht. Einen guten Überblick hierzu gibt <http://www.prostep.org/> sowie http://conferences.esa.int/04c17/SLIDES_PDE2004_Price_exff.ppt

Mit OKSTRA-XML hat auch der OKSTRA® dieser Situation schon Rechnung getragen, und, würde der OKSTRA® heute geschaffen, ist es sehr wahrscheinlich, dass er mit UML modelliert würde und dass als Datenformat der Wahl primär ein XML-basiertes spezifiziert würde.

1.2.2 Stand der Technik: Geometriemodellierung

Auch im Bereich der für den OKSTRA zentral wichtigen Modellierung geometrischer Objekte hat es zwischenzeitlich bedeutsame Initiativen gegeben, getrieben von der internationalen Normung in der ISO und der Standardisierung innerhalb des Open Geospatial Consortium OGC®.

Aus historischen Gründen verwendet der OKSTRA® ein eigenes Geometriemodell (es gab bei seiner Entstehung kein standardisiertes, das für die Aufgabe geeignet gewesen wäre). Mit der Etablierung der ISO 19100-Standards ist hier eine neue Situation eingetreten. Würde man heutzutage ein OKSTRA®-Geometriemodell implementieren, würde sich dies (ganz gleich, ob in EXPRESS oder einer anderen Modellierungssprache) sicher an den ISO-Standards orientieren. Hiermit ist nämlich gewährleistet, dass sich Geodaten des OKSTRA und Geodaten aus anderen Bereichen, die bereits zum ISO-Modell konform sind, problemlos ohne Modelltransformation zusammen verarbeiten lassen.

Für die Geometriemodellierung hat sich eine Arbeitsteilung zwischen ISO und OGC ergeben, wobei die ISO die konzeptionellen Grundlagen standardisiert und die OGC die Implementierungsricht-

linien. Das technische Komitee TC 211 der ISO hat zum Thema Geoinformation einen Standard aus ca. 40 Teilen herausgegeben. Die wichtigsten hiervon betreffen:

- die Repräsentation von Raum- und Zeitbezug (19107, 19108),
- Regeln zur Aufstellung von fachbezogenen Anwendungsschemata (19103, 19109),
- Metadaten (19115, 19139),
- GML und WMS (19136, 19128).

Die vollständige Liste findet sich unter

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=54904&published=on&includesc=true

Die Implementierungsspezifikationen der OGC stützen sich auf das von der ISO standardisierte Modell. Da das OKSTRA-XML als GML-Applikationsschema nicht das OKSTRA[®]-eigene, sondern das ISO-Geometriemodell nutzt, besteht z. B. ein Bruch in der Geometriedarstellung zwischen dem OKSTRA[®]-EXPRESS-Referenzmodell (und damit auch dem OKSTRA-CTE-Format) auf der einen und dem OKSTRA-XML auf der anderen Seite.

1.2.3 Stand der Technik: Referenzierungs- und Identifizierungsmechanismen

Bei der Arbeit mit dem OKSTRA[®] zeigte sich bald, dass es wünschenswert war, Datenbestände schemakonform unter Nichtverletzung von Pflichtrelationen transportieren zu können, wenn die hierdurch referierten Objekte zwar bekannt sind, aber nicht mit ausgetauscht werden sollen. Hierzu wurden als Hilfsmittel neue, zu den Fachobjektarten assoziierte Objektarten, die *symbolischen Verweise* eingeführt. Das OKSTRA Dokument N0015 stellt diesen Sachverhalt mit einem Beispiel versehen genauer dar.

Für die Lösung eines weiteren Identifizierungsproblems sind diese Verweise allerdings nicht geeignet, nämlich die relationale Verknüpfung von Objekten über die Grenzen von Datenbeständen hinweg. Die symbolischen Verweise erlauben nämlich keine Identifizierung von Datenbeständen.

Die Verknüpfung von strukturierten Daten über die Grenzen von physisch zusammengefassten Datenbeständen ist ein zentrales Anliegen der XML-Standardisierung. Diese Spezifikation ist unter dem Namen XLink bekannt. Der XLink-Mechanismus von XML gestattet sowohl, die Aufgabe der symbolischen Verweise zu übernehmen als auch datenbestandsübergreifende Referenzen zu realisieren. Ein xlink wird durch einen URI (Uniform Resource Identifier) realisiert. Es kann sich um dabei eine URL handeln, die den Zugriffsweg zu der referierten Resource (in unserem Falle ein OKSTRA-Objekt) beschreibt, oder aber um eine URN, die die Resource durch einen eindeutigen Namen identifiziert. (Eine ausführlichere Beschreibung gibt der Artikel http://de.wikipedia.org/wiki/Uniform_Resource_Identifier)

Das OKSTRA-XML-Format kann im Gegensatz zu CTE die Vorteile der xlinks nutzen. Möchte man auch im CTE-Format die Möglichkeit externer Referenzen zur Verfügung haben, muss man eine entsprechende Spezifikation für Verknüpfungsobjekte schaffen (die sich ohne weiteres ebenfalls der URIs bedienen könnte) und zusätzlich dafür sorgen, dass diese von CTE-verarbeitender Software auch verstanden wird.

1.2.4 Stand der Technik: Metamodellierung

Zunächst zur Erläuterung: Während die Datenmodellierung Modelle für die Datenobjekte der Anwendungsdomäne spezifiziert, legt die Metamodellierung fest, welche Modellierungselemente dabei verwendet werden dürfen. So werden Modelle u.A. in SQL durch *tables* und *columns* beschrieben, in NIAM durch *entities* und *roles* und in EXPRESS durch *entities* und *attributes*. Bei komplexen Modellen ist es nun oft wünschenswert, das verwendete Metamodell durch eigene Ergänzungen anzureichern. Der OKSTRA liefert mit den bereits erwähnten symbolischen Verweisen ein Beispiel hierfür.

Im Allgemeinen sind Modellierungssprachen nicht durch solche eigenen Modellelemente anreicherbar. In grafisch dargestellten Modellen kann man manchmal durch grafische Mittel, wie die

Farbgebung, Metamodellkonzepte ausdrücken. Für Modelle, die jedoch zur automatischen Nutzung durch Softwarewerkzeuge bestimmt sind, ist dies nutzlos. Hier müssen Metamodellkonzepte durch Konstruktionen innerhalb der Modellierungssprache simuliert werden.

Im Gegensatz zu Techniken wie NIAM, der Entity-Relationship-Modellierung und auch EXPRESS bietet UML verschiedene Mittel, um eigene, anwendungsbezogene Metamodellkonstrukte anzulegen. Ein wichtiges Beispiel für diese Fähigkeit in UML sind die *Stereotype*. Ein UML-Stereotyp ist die Kennzeichnung einer Klasse, dass sie für einen bestimmten Einsatzzweck vorgesehen ist.

Die Verwendung dieser Technik soll hier am Beispiel der symbolischen Verweise gezeigt werden: Hierzu würde man ein Stereotyp `<<symbolisch_vertretbar>>` definieren, das ausdrückt, dass die Instanzen der Klasse sowohl komplette Objekte als auch symbolische Verweise sein dürfen. Das Stereotyp ist Bestandteil des OKSTRA-Modells und muss zusätzlich durch entsprechende Dokumentation erläutert werden, die sagt, was es in der Praxis bedeutet. Man beachte, dass durch die Nutzung dieses Mechanismus für jede symbolisch vertretbare Objektart in UML nur eine Klasse hingeschrieben werden muss.

1.2.5 Folgerungen

Auch wenn die Modellierung mit NIAM und EXPRESS sich bewährt hat, so zeigen die in den vorigen Abschnitten angestellten Überlegungen, dass die OKSTRA-Modellierung gegenüber den bei Standardisierungsprozessen für Datenmodelle heutzutage bevorzugten Verfahrensweisen einen Sonderweg geht. Aus diesem Grunde ist der OKSTRA schon früh in Form seiner XML-Variante dem Stande der Technik angenähert worden, was z.B. eine standardisierte Realisierung von Geo-Web-Services für den OKSTRA erst ermöglicht.

Die vorliegende Machbarkeitsstudie untersucht, welche Auswirkungen zu erwarten sind, wenn dieser Prozess der Verwendung heutzutage etablierter Praktiken bei der Datenmodellierung konsequent weitergeführt wird. Das würde bedeuten, dass die Referenzmodellierung von EXPRESS auf UML umgestellt würde und daraus über die in den ISO-191xx-Standards festgelegten Prozeduren GML-Applikationsschemata erzeugt werden. Es bedeutet auch, dass das neue OKSTRA-XML die gegenüber CTE bevorzugte Kodierung für Datenbestände wird.

Durch diese Studie sind hierzu folgende Fragen zu klären:

- Wie wird sichergestellt, dass existierende OKSTRA[®]-Versionen und zugehörige Datenbestände weiterhin nutzbar bleiben?
- Welche Änderungen ergeben sich für die Entwickler von SQL-Datenbanken (z.B. Hessen-SIB), die bisher das EXPRESS-Schema individuell umgesetzt haben?
- Bestehende Schemata müssen von EXPRESS in UML umgestellt werden, was wegen der Menge des Materials nur mit massiver automatischer Unterstützung realistisch ist. Welche Lösungsmöglichkeiten gibt es hierfür?
- Welche Gremien und Personen sind einzubeziehen, um eine ausreichende Akzeptanz des neuen Vorgehens bei der OKSTRA[®]-Community sicherzustellen?
- Die vorhandenen Code-Generierungsaufgaben (Dokumentation, OKSTRA[®]-Wiki, OKSTRA[®]-Prüfwerkzeug) müssen weiter erfüllt werden. Welche Lösungsmöglichkeiten existieren?
- Müssen EXPRESS-Modelle aus der neuen Referenzmodellierung abgeleitet werden? Wenn ja, über wie viele Versionen hinweg?
- Wie sind die Prozesse der Pflegestelle und bei Herstellern und Anwendern betroffen? Ergeben sich Rationalisierungseffekte und ist Investitionsschutz hinreichend sicherzustellen?
- Welche Produkte sollen von der Pflegestelle im Vergleich zu vorher angeboten werden?
- Welche Grundlagen-Modelle müssen überarbeitet werden?
- Welche technischen Regeln, z.B. Benennungsregeln für Objektarten und Attribute, sind betroffen?

2 Technische Umsetzbarkeit

Bei einer Ablösung der bisherigen Referenzmodellierung in EXPRESS durch eine solche in UML ist sicherzustellen, dass alles, was heutzutage in jener Modellierung ausgedrückt ist, auch in der neuen formuliert werden kann. Deshalb muss geklärt werden, ob und wie die beim OKSTRA® genutzten Sprechweisen aus EXPRESS und die eigenen Metamodellzusätze in UML ausdrückbar sind.

Dieses Kapitel untersucht diesen Punkt zuerst und widmet sich dann der Frage, ob das Vorhaben beim jetzigen Umfang des OKSTRA® realistisch ist, und wenn dies bejaht wird, wie dabei vorzugehen ist.

2.1 *Abbildbarkeit des OKSTRA-EXPRESS-Modells nach UML*

Dieser Abschnitt nennt die zz. im OKSTRA verwendeten Modellkonstrukte aus EXPRESS und dem OKSTRA-eigenen Metamodell. Dieses wurde in Version 1.012 formalisiert und ist im OKSTRA-Dokument N0092 dokumentiert.

Im Folgenden wird häufig Bezug genommen auf die *Stereotype* in UML. Dies sind Metamodellelemente zur Erweiterung von UML nach eigenen, anwendungsbezogenen Regeln. Eine detaillierte Erläuterung findet sich unter: [http://de.wikipedia.org/wiki/Stereotyp_\(UML\)](http://de.wikipedia.org/wiki/Stereotyp_(UML))

2.1.1 Schemata

Der zum Begriff des EXPRESS-Schema äquivalente Begriff ist das *Paket (package)*. Pakete bieten in UML zusätzlich die Trennung der Namensräume für die in den Paketen enthaltenen Modellobjekte.

2.1.2 Entities und Vererbung

Der zum Begriff des EXPRESS-Entity äquivalente Begriff ist die *Klasse*. Wie in EXPRESS können Klassen in Vererbungsbeziehungen stehen, eine Klasse kann Eigenschaften von mehreren Oberklassen erben, und es gibt den Unterschied zwischen abstrakten und konkreten Klassen. Nur von letzteren können Objektinstanzen hergestellt werden.

Das OKSTRA-Metamodell unterscheidet klassifiziert die Entities zusätzlich in

- vollwertige Objektarten
- konzeptionelle Objektarten
- Schlüsseltabellen
- Hilfsobjektarten für symbolische Verweise.

Konzeptionelle Objekte sind nur innerhalb anderer Objekte lebensfähig, d.h. sie können nicht außerhalb eine enthaltenden Objektes instanziiert werden. Dieses Konzept kann in UML z.B. durch die Kompositionsbeziehung (eine spezielle Art der Assoziation) ausgedrückt werden. Zusätzlich können diese Objektarten auch mit dem UML Stereotyp <<datatype>> versehen werden, dass die Besonderheit dieser Objektarten sehr gut abbildet (siehe z.B. [Oes04]).

Die Schlüsseltabellen und die Hilfsobjekte stellen Sonderfälle dar, die unten in eigenen Unterabschnitten abgehandelt werden.

2.1.3 Attribute

UML gestattet wie EXPRESS die Spezifikation getypter, ein- und mehrwertiger Attribute in Klassen.

2.1.4 Relationen

Relationale Beziehungen zwischen Objekten werden in UML durch *Assoziationen* ausgedrückt, die zwischen Objektarten bestehen. Alternativ ist auch eine Darstellung durch objektwertige Attribute möglich (analog zu EXPRESS), dann entfällt jedoch die grafische Repräsentation des Bezugs.

Assoziationen verbinden Objektarten über die *Assoziationsenden*, die Eigenschaften zur näheren Charakterisierung des Bezuges aufweisen.

Die *Navigierbarkeit* einer Relation wird in UML durch eine entsprechende Eigenschaft des Assoziationsendes ausgedrückt

Die *Kardinalitäten* einer Relation werden in UML durch Multiplizitäten von Assoziationsenden wiedergegeben.

Die oben bei den konzeptionellen Objektarten erwähnte *Komposition* ist eine spezialisierte Form der Assoziation.

Für die Abbildung *objektifizierter Relationen* stellt UML ein eigenes Konstrukt bereit, die Assoziationsklassen.

Ist eine Relation *geordnet* (EXPRESS: LIST OF), so kann dies in UML ebenfalls über eine Eigenschaft des Assoziationsendes ausgedrückt werden.

2.1.5 Datentypen

Zur Abbildung der TYPEs von EXPRESS sind in UML die *Datatypes* geeignet, aus denen ebenfalls Typhierarchien gebildet werden können. Sie werden als Klassen mit dem Stereotyp <<datatype>> gekennzeichnet. Die zu einem Datatype gehörenden Objektinstanzen können nicht außerhalb anderer Objekte existieren. Beispiel: Länge mit den Attributen: Maß (muss > 0 sein), Maßeinheit. Über Datatypes kann auch das bisherige im OKSTRA verwendete Definitionsverfahren 1:1 übernommen werden.

2.1.6 Zwangsbedingungen (Constraints)

Zur Formulierung von Bedingungen steht in UML eine eigene Sprache zur Verfügung, die *Object Constraint Language* (OCL). OCL-Ausdrücke werden an die UML-Modellobjekte gehängt. Das Modellobjekt definiert den Kontext für den OCL-Ausdruck. Z.B. kann eine OCL-Bedingung im Kontext einer bestimmten Klasse Einschränkungen für Attribute definieren. (Eine ausführlichere Erläuterung mit Beispielen findet sich in <http://de.wikipedia.org/wiki/OCL>)

2.1.7 Schlüsseltabellen

Das Konzept der Schlüsseltabelle im OKSTRA ist weder in NIAM, EXPRESS oder UML leicht abbildbar. Daher verpackt die EXPRESS-Modellierung die Spezifikation von Schlüsseltabellen auch in Kommentaren mit spezieller Syntax. Nach vorliegender Einschätzung existieren verschiedene Möglichkeiten zur Behandlung in UML mit unterschiedlichen Vorzügen und Nachteilen (etwa Verlagerung in externe Ressourcen, Verlagerung in die Kommentierung, Nutzung von OCL; diese Liste erhebt keinen Anspruch auf Vollständigkeit). Diese Möglichkeiten wären im Detail zu untersuchen und auszuprobieren.

2.1.8 Symbolische Verweise

Das Konzept der symbolischen Verweise verlangt, dass als Ziel einer Relation sowohl die referierten Objekte als auch die sie vertretenden symbolischen Verweise vorkommen dürfen. Das ist in EXPRESS nur durch die Einführung eines weiteren Supertypen für *jede* symbolisch vertretbare Objektart möglich, von dem dann diese Objektart und die entsprechende Objektart für die Verweise erbt. Es entstehen also für jede symbolische vertretbare Objektart insgesamt drei EXPRESS-Entities.

In UML können Objektarten, deren Instanzen durch symbolische Verweise vertreten sein können, z.B. durch ein entsprechendes Stereotyp gekennzeichnet werden. Dabei ist z.B. festzulegen, welches der Attribute der Objektart das Vertretungssymbol sein soll (z.B. über den Namen *symbol*). Es sind aber auch mehr UML-intrinsische Abbildungsmuster denkbar, z.B. durch Nutzung von *unions*. (Die Attribute einer als *union* gekennzeichneten Klasse sind nicht gleichzeitig, sondern wahlweise in den Objektinstanzen repräsentiert, ähnlich *choice* in XML Schema)

2.1.9 OKSTRA-Id

Die OKSTRA-Id wird zz. durch eine abstrakte Oberklasse eingeführt, die an der Wurzel der Hierarchie der Objektarten steht. Diese Vorgehensweise kann und sollte beibehalten werden.

2.1.10 Fazit

Modellierungstechnisch ist die Nutzung von UML für den OKSTRA möglich und bringt an einigen Stellen auch Vorteile durch straffere und weniger komplexe Formulierungen. Insbesondere kann das OKSTRA-eigene Metamodell weitestgehend selbst mit UML-Mitteln ausgedrückt werden. Ein Problemfall bleiben, wie in EXPRESS, die Schlüsseltabellen.

Alle vereinbarten Regeln für die Verwendung von UML für die OKSTRA-Modellierung sind in einem *UML-Profil* niederzulegen und zu dokumentieren. Zum Prozess der Profilerstellung siehe [http://de.wikipedia.org/wiki/Profil_\(UML\)](http://de.wikipedia.org/wiki/Profil_(UML))

2.2 Umwandlung des Express-Codes in UML

Eine Möglichkeit zur Gewinnung des UML-Modells aus EXPRESS besteht in der händischen Umsetzung, wobei gemäß den EXPRESS-Schemata mit einem UML-Designwerkzeug die OKSTRA-Schemata neu erschaffen werden. Beim derzeitigen Umfang des OKSTRA von ca. 1500 Entities ist dieses Verfahren jedoch wirtschaftlich nicht vertretbar. Eine eher knapp kalkulierte Bearbeitungszeit von 0.5 h pro Entity würde einen Aufwand von ca. 4-5 Personenmonaten erzeugen, was ca. 85% des derzeitigen OKSTRA-Pflegebudgets pro Jahr entspricht.

Es ist daher zu untersuchen, ob eine weitgehende Automatisierung des Prozesses möglich ist. Aus den im ersten Kapitel genannten Gründen ist der OKSTRA nicht der erste Anwendungsfall dieser Art. Andere Autoren haben sich dem Problem ebenfalls gestellt. Vorhandene Ansätze finden sich etwa unter:

<http://nfig.hd.free.fr/util/Referential/Ontology/Ontology%20Definition%20Metamodel/MEXICO%20V1%20UJ.ppt>

sowie

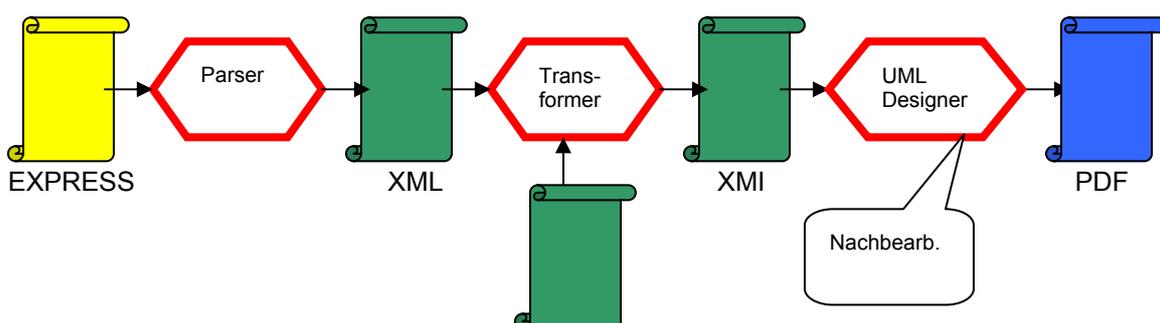
<http://exff.org/> mit Software unter <http://exff.sourceforge.net/>

Insbesondere dieser letzte Ansatz zeigt die grundsätzliche technische Umsetzbarkeit.

Das primäre Ziel ist die Umwandlung der EXPRESS-Schemata in eine Form, die sich leicht in UML umwandeln lässt. Mit Hilfe von XMI (XML Metadata Interchange, eine Sprache zur Übertragung von Modellen mit spezifizierbarem Metamodell) steht so eine Möglichkeit zur Verfügung, d.h. ein Softwarewerkzeug setzt die EXPRESS-Schemata in äquivalentes XMI um, und dieses wird in ein gängiges UML-Designwerkzeug importiert. Das genannte exff-Projekt beschreitet genau diesen Weg und hat dabei noch den Vorteil, eine Open-Source-Lösung anzubieten.

Es wurden bei den Autoren auf diese Art testweise das Straßennetzmodell, die Geometrie, die Historisierung und die Allgemeinen Objekte umgesetzt. Das dabei verwendete UML-Designwerkzeug „Enterprise Architect“ wird wegen seines günstigen Preises bei enormem Funktionsumfang oft in UML-Projekten eingesetzt, siehe <http://www.sparxsystems.de/default.asp?nav=3x11&lid=29>.

Die folgende Grafik erläutert das Vorgehen:



XSLT

Wie man sieht, verläuft die Umsetzung zum XMI zweistufig, in dem zuerst eine dem EXPRESS genau entsprechende XML-Zwischenstufe erzeugt wird. Die eigentliche Umsetzung wird durch das XSLT-Stylesheet gesteuert. Hier hat man die Möglichkeit, durch Ergänzen des Stylesheets weitere Modellregeln zu implementieren.

Folgende Umsetzungsschritte erfolgten unter Verwendung des XSLT-Stylesheets aus dem exff-Projekt automatisiert:

- Erzeugung von UML-Classes aus den EXPRESS-Entities,
- Erzeugung von UML-Generalisierungen aus den EXPRESS-Vererbungsbeziehungen,
- Umsetzung von primären Relationen in UML-Assoziationen,
- Umsetzung von Attributen mit EXPRESS-Basisdatentypen.

Folgendes leistet das XSLT-Stylesheet nicht:

- Umsetzung des OKSTRA[®]-Metamodells in geeignete Stereotypes,
- Abbildung von abgeleiteten Datentypen und Typhierarchien,
- Abbildung von inversen Relationen,
- Abbildung von Zwangsbedingungen (WHERE-Clauses),
- Abbildung der Wertekataloge von Schlüssel Tabellen,
- Ersetzung von symbolischen Verweisen durch normale Assoziationen.

Bei der testweisen Umsetzung der Autoren wurden in einem manuellen Vorverarbeitungsschritt im umzuwandelnden EXPRESS-Schema abgeleitete Datentypen durch die entsprechenden EXPRESS-Basistypen und symbolische Verweise durch normale Relationen ersetzt. Im automatisiert erzeugten UML-Modell wurden die inversen Relationen zu den umgewandelten primären Relationen händisch ergänzt. Durch eine Erweiterung des bei der Umsetzung verwendeten XSLT-Stylesheets wäre es möglich, auch die inversen Relationen (und ggf. noch weitere Modellelemente) automatisiert umzusetzen.

Neben diesem prototypisch ausgewerteten Verfahren sind auch andere Ansätze denkbar. So könnten Werkzeuge, die zur Generierung von Klassenbibliotheken oder Dokumentation aus den EXPRESS-Schemata verwendet werden, so erweitert werden, dass sie zusätzlich das gewünschte XMI-Format erzeugen. Dieser Ansatz böte den Vorteil, dass die beim obigen Verfahren nicht durchgeführten Operationen ebenfalls automatisiert werden könnten, soweit sich hierfür algorithmisch umsetzbare Regeln finden lassen.

Unabhängig vom gewählten Verfahren ist der Umsetzung das notwendige UML-Profil zugrunde zu legen (siehe 2.1.10).

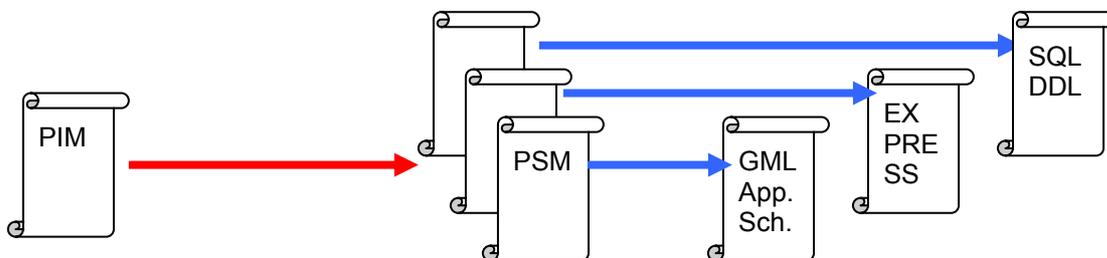
2.3 GML und der MDA-Ansatz

Aus dem UML-Schema muss nun das GML-Applikationsschema gewonnen werden. Grundsätzlich wird dieser Schritt durch ISO 19136 Annex E geregelt. Hierbei bestehen jedoch Restriktionen, denen das EXPRESS-Modell und das daraus gewonnene UML-Modell nicht unterliegen. So ist z.B. nur einfache Vererbung zugelassen. Das Problem ist nicht neu und trat bereits bei der Formulierung des heutigen OKSTRA-XML auf.

Hat man erst einmal ein Annex E-konformes Modell gewonnen, kann die Umsetzung in das entsprechende GML-Applikationsschema wiederum automatisch erfolgen. Hierzu gibt es das Open-Source-Werkzeug Shape Change (<http://www.interactive-instruments.de/ugas/>).

Um das Problem des nicht Annex-E-konformen UML zu lösen, greift man auf eine Basistechnik aus dem Bereich der *Model Driven Architecture* (MDA) zurück. Man betrachtet das Referenzmodell als plattform-unabhängiges Modell (PIM) und leitet daraus durch Modelltransformationen plattform-

spezifische Modelle (PSM) ab. Das Modell gemäß Annex E wäre demzufolge ein solches PSM. Man beachte, dass die PSMs ebenfalls in UML/XMI vorliegen und visualisiert werden können (und zur Not nachbearbeitet). Durch einen letzten Schritt wird aus dem PSM schließlich Code, z.B. XML-Schema. Die folgende Grafik illustriert dies.



Damit der MDA-Ansatz funktionieren kann, muss auch das PIM in einem strikt verbindlichen UML verfasst sein, weil sonst automatische Modelltransformationen nicht realisierbar sind. Das bedeutet, dass für den OKSTRA ein UML-Profil spezifiziert werden muss, das sagt, welche UML-Konstrukte verwendet werden dürfen und insbesondere, welche Ergänzungen, z.B. Stereotype, zulässig sind.

Für jedes benötigte PSM sind sodann die Transformationsregeln zu entwickeln und umzusetzen. Für nicht zu komplexe Modelltransformationen können diese etwa als XSLT-Stylesheets, die XML in XML umsetzen, realisiert werden.

Bei eingehenderer Betrachtung bietet der MDA-Ansatz noch weitere Möglichkeiten:

- EXPRESS-Modelle könnten weiter unterstützt werden. Dazu müsste ein weiteres PSM in UML automatisch erzeugt werden, das sich leicht in EXPRESS umsetzen lässt. Man benötigt hierzu eine EXPRESS-Metamodell, das zusätzlich um die OKSTRA-Metamodell-Ergänzungen und die in Kommentaren untergebrachten Generierungsanweisungen zu erweitern ist. EXPRESS-Metamodelle gibt es bereits aus dem exff-Projekt, diese könnten evtl. als Ausgangspunkt dienen.
- Die Unterstützung von SQL könnte verbessert werden. Die SQL-Modellierung des OKSTRA wurde seinerzeit wegen des geringen Nutzens der Modelle eingestellt. Stattdessen war die Anwendergemeinschaft gebeten worden, praxistaugliche OKSTRA-SQL-Modelle auf den OKSTRA-Webseiten zu publizieren. Mit Hilfe von PSMs für die Umsetzung in SQL-DDL könnten evtl. praxiserprobte Techniken zum Aufbau performanter relationaler Datenbanken formalisiert werden. Allerdings ist die Spezifikation derartiger PSMs sehr anwendungsabhängig, und vermutlich sind für unterschiedliche Datenbankfabrikate auch unterschiedliche SQL-Codegeneratoren erforderlich. Eine allgemeingültige Lösung ist daher auch mit dem MDA-Ansatz unrealistisch.
- Schließlich kann die Pflege von Klassenbibliotheken mit Hilfe von PSMs rationalisiert werden.

2.4 Überarbeitung von Basismodellen

Die Nutzung des Geometriemodells in OKSTRA-XML zeigt, dass eine Ablösung des bisherigen Geometriemodells nicht nur wünschenswert ist, sondern auch durchführbar.

Die Überarbeitung weiterer Basismodelle wird nicht als vordringlich eingestuft. Sollte Bedarf hierfür bestehen, kann sie später erfolgen.

2.5 Zu modifizierende lexikalische und syntaktische Regeln

Durch den MDA-Ansatz besteht weitestgehende Freiheit für die Bezeichnung von Klassen, Attributen und Assoziationsrollen in UML. Etwaige plattformspezifische Bezeichnungsregeln müssen bei

der Transformation des PIM in die PSMs durch evtl. notwendige Umbenennung berücksichtigt werden.

3 Auswirkungen der Umstellung auf Prozesse

3.1 Auswirkungen auf die Pflegestelle

3.1.1 Bisheriger Pflegeprozess

Der Pflegeprozess besteht kurzgefasst aus folgenden Schritten:

1. Eingang eines Änderungsantrages registrieren, Antrag vorprüfen, Annahme bestätigen oder Ablehnung mitteilen
2. Bei Annahme Erarbeitung einer Reaktion (z.B. betroffene Schemata identifizieren, weitere Vorgehensweise bestimmen)
3. Nach Freigabe zu Änderung fachliche Voraussetzungen berücksichtigen (Regelwerke, Expertengruppen)
4. Resultate in Modellierungsvorschlag in NIAM plus evtl. EXPRESS umsetzen. Verfeinerung des Modellierungsvorschlages durch Wiederholung von Schritt 3 und 4
5. Bei nächster Versionierung die aufgelaufenen Änderungen in das bestehende Modell integrieren (NIAM, EXPRESS)
6. Neue Version als Kandidat veröffentlichen
7. Evtl. technische Korrekturen vornehmen
8. Weitere Produkte herstellen (OKSTRA-XML, Dokumentation, Wikiseiten) und veröffentlichen

Von einer Umstellung der Referenzmodellierung sind betroffen die Schritte 4 bis 8.

3.1.2 Erwartete Änderungen

Die NIAM-Modellierung sollte als Ausgangspunkt zunächst erhalten bleiben, die Akzeptanz bei Experten und Anwendern ist hoch. In früheren Forschungsarbeiten (Stichwort Objektorientierter OKSTRA, siehe Dokument N0048) hat sich gezeigt, dass UML-Diagrammen nicht unbedingt geeignet sind, um mit UML-unkundigen Fachexperten kurzfristig Objekte zu diskutieren. Wie bereits dargelegt ist einer der Vorteile von NIAM die geringe Zahl benötigter und damit zu erlernender Modellelemente. NIAM würde im wesentlichen als Mittel zur kooperativen Erarbeitung neuer oder zu ergänzender Modelle eingesetzt werden.

Das Referenzmodell mit normativem Charakter würde daraus und aus den informellen Ergänzungen zu den NIAM-Diagrammen in UML erstellt. Hierbei ist das noch zu erarbeitenden UML-Profil zu verwenden. Die linearisierte Form dieses Referenzmodells in XMI (XML Metadata Interchange, eine Sprache zum Transport von Modellen) dient dann als Ausgangspunkt für alle Produkte, die zur Nutzung des OKSTRA benötigt werden.

Hierzu ist zunächst anzumerken, dass die NIAM-Modelle ihre Wichtigkeit für das Verständnis der Modelle verlieren dürften. Denn weil die *komplette* Referenzmodellierung dann grafisch in Form von UML-Diagramme vorliegt, ist die Schwelle für den Zugang dazu niedriger als bei EXPRESS. Auch heute reichen die NIAM-Diagramme nicht aus, wenn es um die Recherche nach Attributen sowie ihre Festlegung in den Expertengruppen geht. Es ist daher anzustreben, dass die Pflege der NIAM-Diagramme zu Zwecken der Dokumentation künftig entfallen kann.

Für die Pflege des UML-Modells muss ein UML-Designtool ausgewählt werden. Wie bereits festgestellt, bietet sich der Enterprise Architect hierzu aus Preis- und Leistungsgründen an.

Die Arbeit der Pflegestelle wird sich insofern verändern, als die separate parallele Pflege von NIAM, EXPRESS und Dokumentation wegfällt. Stattdessen werden grafische und lineare Repräsentierung sowie die Dokumentation mit dem UML-Designtool gemeinsam gepflegt.

Ähnlich wie es heute schon für OKSTRA-XML der Fall ist, wird ein Generierungswerkzeug für die PSMs und die Codegenerierung (z.B. GML) benötigt. Hier ist ein Umbau des vorhandenen Werkzeugs als technisch realistisch anzusehen.

Für die Umstellung ist außerdem eine einmalige Überarbeitung der OKSTRA-Web-Seite erforderlich.

3.2 Auswirkungen auf die Softwarehersteller

Da eine große Anzahl von UML-Tools (tlw. kostenlos bzw. kostengünstig) verfügbar sind und UML verbreitet zur Spezifikation von Software eingesetzt wird, sehen wir die Nutzbarkeit des OKSTRA durch Hersteller, die sich neu mit OKSTRA-Technologie beschäftigen wollen, eher unterstützt.

Eine Umstellung des Modellgenerierungsprozesses in der beschriebenen Art hat jedoch evtl. zur Folge, dass sie so generierten Schemata zwar dasselbe Datenmodell wiedergeben, sich formal von den heutigen unterscheiden. Dies gilt ziemlich sicher für die GML-Variante, möglicherweise aber auch für eine evtl. weiterhin bereitgestellte EXPRESS-Variante.

Bei den Herstellern, die bisher schon OKSTRA-konforme Produkte anbieten, würden dehalb ggf. Umstellungsaufwände anfallen. In welcher Größenordnung, hängt davon ab, ob EXPRESS weiterhin unterstützt würde und wie sehr die umgestellten Modelle und XML-Schemata von den bisherigen Versionen abweichen. Welchen Umfang diese Auswirkungen auf die Entwicklungsprozesse bei den Herstellern im Detail haben werden, ist nicht abzuschätzen, da nicht bekannt ist, mit welchen Vorgehensmodellen dort gearbeitet wird.

Ob die UML-Umstellung in Bezug auf SQL Vorteile für die Hersteller bietet, ist ungewiss und müsste mit interessierten Herstellern diskutiert werden. Grundsätzlich könnte, wie dargelegt, auch SQL-DDL aus einem entsprechenden PSM gewonnen werden. Allerdings sind, wie weiter oben dargelegt, die Anforderungen an die Datenbankarchitektur anwendungsabhängig und nicht standardisierbar, so dass die Hersteller selbst für den Aufbau der benötigten PSMs verantwortlich wären.

3.3 Auswirkungen auf die Anwender in Bund und Ländern

Da die Referenzmodellierung nicht mehr in EXPRESS zur Verfügung steht, wird auch die jetzige Dokumentation EXPRESS-HTML hinfällig. Die Anwender müssen sich also auf eine anders aufgebaute Dokumentation einstellen. Insbesondere wird es eine grafische Darstellung des Modells in Form von UML-Klassendiagrammen geben, so dass Grundkenntnisse zum Verständnis solcher Diagramme benötigt werden. Das trifft auch auf die Fachexperten zu, sobald Modelle im Detail betrachtet werden müssen. Hier wird die Pflegestelle Handreichungen mit Erläuterungen und Beispielen schaffen müssen.

4 Auswirkungen der Umstellung auf Produkte

Zurzeit werden die abgeleiteten Produkte zum OKSTRA aus den EXPRESS-Schemata gewonnen. Dazu sind auch zusätzliche in Kommentaren kodierte Anweisungen für den Generierungsprozess vorgesehen. Grundsätzlich sind zukünftig zwei Strategien denkbar:

1. Bei einer Umstellung der Referenzmodellierung auf UML könnte der Generierungsprozess für einige Produkte beibehalten werden, wenn EXPRESS weiter unterstützt wird und die Generierungshilfen irgendwie durch den oben geschilderten Modellgewinnungsprozess hindurchgeschleust werden könnten. Das hätte den Vorteil, dass die Generierungswerkzeuge weiter nutzbar wären.
2. Alle Produkte werden einheitlich über den MDA-Ansatz erzeugt.

Im Folgenden werden die betroffenen Produkte einzeln betrachtet.

4.1 Schemata

4.1.1 Referenzmodell

Das Referenzmodell wird standardmäßig als XMI-Datei aus dem UML-Designwerkzeug produziert und kann auch als solches publiziert werden.

4.1.2 Plattform-spezifische Schemata

Sie werden während des Modellgewinnungsprozesses generiert und bräuchten nicht publiziert werden. Wir sehen in der Bereitstellung dieser Schemata keinen Zusatznutzen für die Nutzer des OKSTRA.

4.1.3 GML-Applikationsschema

Für eine automatische Erzeugung der GML-Applikationsschemata könnte weiterhin der bisherige Weg genutzt werden, falls EXPRESS weiter unterstützt wird, oder die Schemata könnten in konsequenter Anwendung des MDA-Ansatzes aus einem entsprechenden UML-PSM z.B. mit Hilfe von ShapeChange nach den Vorgaben von ISO19136 Annex E gewonnen werden.

U.U. ist es pragmatisch der beste Weg, den bisherigen Generierungsprozess und die Unterstützung von EXPRESS zunächst beizubehalten und eine weitergehende Umstellung erst bei Bedarf zu realisieren.

4.1.4 EXPRESS-Schemata

Die EXPRESS-Schemata würden, falls gewünscht, aus dem Referenzmodell über ein entsprechendes PSM erzeugt.

4.1.5 SQL-Schemata

Die Gewinnung von SQL-Schemata obliegt den Herstellern der gewünschten Datenbankanwendungen.

4.2 Dokumentation

Bei Umstieg auf UML wird eine integrierte Dokumentation (Diagramme + Erläuterungen) automatisiert mit Hilfe des UML-Designwerkzeugs abgeleitet. Die Erläuterungen werden den Elementen des formalen Referenzmodells zugeordnet und sind hierüber auch navigierbar. Eine webfähige Dokumentation kann mit Hilfe des Designwerkzeugs erzeugt werden, so dass ein separater Dokumentationsgenerator a la EXPRESS-HTML an sich nicht mehr benötigt würde. Allerdings müsste dazu untersucht werden, ob die derzeit gegebene Möglichkeit des objektartweisen Versionsvergleichs weiterhin unterstützt werden soll, und wenn ja, welche Möglichkeiten es hierfür gibt.

Die bisherige EXPRESS-HTML-Dokumentation kann bei Beibehaltung der EXPRESS-Version weitergepflegt werden, solange EXPRESS noch unterstützt werden soll und eine solche Form der Dokumentation von den Nutzern gewünscht wird.

4.3 Wiki

Es ist fraglich, ob das Wiki angesichts der Möglichkeit, Dokumentation mit Zusatzerläuterungen direkt aus dem UML-Tool zu erzeugen, noch einen ausreichenden zusätzlichen Nutzen hat. Der Hauptnutzen eines Wiki besteht in der Möglichkeit der kooperativen Erfassung von Einträgen; dies wurde jedoch bisher nur unzureichend genutzt (Kritische Masse an Verfassern von Beiträgen ist zu gering).

Der Generierungsprozess für die Wiki-Seiten basiert momentan auf der Basis von EXPRESS. Sollte ein Wiki weiterhin erwünscht sein, müssten die Wiki-Seiten mit Hilfe des MDA-Ansatzes erzeugt werden, da schließlich das UML-Referenzmodell damit dokumentiert werden soll. Dann muss aber auch Sorge dafür getragen werden, dass Änderungen in den Wiki-Seiten, die durch deren Nutzer vorgenommen werden, wieder (nach Qualitätskontrolle) in das Modell zurückfließen können.

4.4 Programm-Bibliotheken

Der Generierungsprozess für die Klassenbibliothek XTRA kann sowohl über EXPRESS, wenn es weiter unterstützt wird, als auch aus der aus dem UML-Modell exportierten XMI abgeleitet werden. Hierzu müsste der Generator entsprechend umgebaut werden. Zusätzlich wären auch die XML-Import- und Export-Module anzupassen, um erwartete strukturelle Veränderungen im OKSTRA-XML durch den veränderten Generierungsprozess und den Austausch des Geometriemodells zu akzeptieren.

Analoges gilt für die offene OKSTRA-Klassenbibliothek.

In jedem Fall wäre die betroffene Bibliothek neu zu generieren.

4.5 Prüftool

Das Prüftool verwendet zurzeit die XTRA-Klassenbibliothek und ist deshalb nur mittelbar betroffen. Es müsste mit der gemäß 4.4 neu erstellten Bibliotheksversion neu gebunden werden. Dies gilt auch, wenn eine Umstellung auf die Klassenbibliothek erfolgt.

4.6 Normausgabetool

Entsprechendes zu 4.5 gilt für dieses Produkt.

4.7 OKSTRA-konforme Web Services

Von den im Dokument N0112 aufgeführten Maßnahmen zur Unterstützung von OKSTRA-konformen Web Services betrifft nur die „Normalisierung des OKSTRA“ (Abschnitt 7.6 in N0112) den Modellierungsprozess. Hier sollte zunächst geprüft werden, in wie weit diese Anforderungen Auswirkungen auf das OKSTRA-Metamodell haben. Etwaige hinzukommende neue Anforderungen sind dann in das hierfür zu erstellende UML-Profil zu integrieren.

5 Migrationskonzept

5.1 *Erhaltung der Kompatibilität mit alten Versionen*

Alte Versionen werden weiter vorgehalten, Umstellung erfolgt durch Einführung einer neuen Version. Diese neue Version sollte ausschließlich die Umstellung auf UML beinhalten und keine weiteren fachlichen Versionierungen, damit die Betroffenen die Möglichkeit haben, während ihrer Umstellungsphase sowohl eine aktuelle EXPRESS- als auch die neue UML-Version nutzen zu können.

5.2 *Akzeptanz bei den Betroffenen*

Zunächst ist die breite Unterstützung durch die PG OKSTRA sicherzustellen. Es sollte dann ein Meinungsbild bei möglichst allen Betroffenen eingeholt und ausgewertet werden. Damit dies funktioniert, müsste auf der OKSTRA-Webseite ein Prototyp für die neue Modellierung installiert werden. Neben den UML-Diagrammen muss auch die neue integrierte Dokumentation hierbei gezeigt werden.

5.3 *Arbeitsschritte für die Migration*

Folgende Arbeitsschritte werden notwendig sein:

- UML-Profil aus dem OKSTRA-Metamodell entwickeln
- Entwicklung von Werkzeugen, die EXPRESS verarbeiten, um die benötigten XML-Dokumente damit zu erstellen. Statt einer Neuentwicklung ist hierbei möglichst die Erweiterung existierender Werkzeuge vorzuziehen.
- Umsetzung der Schemata. Hier sind evtl. noch manuelle Vorbereitungen in den EXPRESS-Schemata notwendig und/oder Nachbereitungen in den generierten UML-Schemata. Die manuellen Eingriffe sollen durch den vorherigen Verfahrensschritt minimiert werden.
- Erzeugung der Diagramme mit dem UML-Designwerkzeug.
- Dokumentation mit dem UML-Designwerkzeug einbringen bzw. ergänzen.

Hinweis: Die Nutzung zusätzlicher, wünschenswerter UML-Techniken (z.B. Aggregation, Komposition) sollte bei der Erstumstellung nicht berücksichtigt werden. Hierfür wären nämlich zunächst semantische Regeln zu erstellen. Die Berücksichtigung solcher Regeln kann ohne weiteres später während des regulären Pflegeprozesses erfolgen, und zwar dann, wenn ein Schema sowieso zu überarbeiten ist.

Das durchgeführte Prototyping hat gezeigt, dass ein bereichsweises Vorgehen bei der Migration (also z.B. zuerst nur Netz und Bestand) nicht durchführbar ist, denn der OKSTRA ist in sich so dicht verflochten, dass die Migration eines Bereiches die Migration aller damit zwangsweise zusammenhängenden Bereiche erfordert.

6 Resümee

Als Ergebnis dieser Studie ist festzuhalten:

Eine Umstellung der OKSTRA-Referenzmodellierung auf UML als Modellierungssprache ist als technisch durchführbar anzusehen. Der Problempunkt Schlüsseltabellen ist zwar noch gesondert zu untersuchen, wird aber als lösbar betrachtet.

Die Arbeit der OKSTRA-Pflege wird weiterhin im Wesentlichen durch die Moderation der Experten erbracht, sodass sich die Einsparungen bei der OKSTRA-Pflege durch die Umstellung nach UML in Grenzen halten werden. Der weitaus größere Nutzen und dementsprechend größere Kostenvorteil wird sich bei der Verbindung der Klassenbibliothek mit einem UML-Generator ergeben (siehe Darstellung im Grobkonzept zur Klassenbibliothek).

Voraussetzung für eine Verwendung eines UML-Modells als Referenz, aus der alle benötigten Produkte erzeugt werden, muss eine breit gefächerte Akzeptanz bei Herstellern und Anwendern und last not least auch bei den in der OKSTRA-Pflege mitwirkenden Experten sein. Zum Erreichen dieser Akzeptanz

- sind vorbereitende und begleitende Maßnahmen erforderlich,
- müssen die existierenden Versionen des OKSTRA weiter in unveränderter Form bereitgestellt werden
- können bei Bedarf auch weiterhin für eine noch näher zu bestimmde Übergangszeit EXPRESS-Modelle als abgeleitete Modelle bereitgestellt werden.