

Nr.	Thema	Status	Bemerkungen	Wo?
6	Der Typ von "TAN-Listenstatus" in "TANListDisplay-1" muss "code" statt "an" sein.	implementiert		Schema (TANListDisplay-1.xsd): geändert
9	In ResponseState muss die Restriktion für den RespState um die Info-Codes erweitert werden ([0139]...).	implementiert		Schema (common.xsd): geändert
11	In Subscription_1_Req sind die Werte für Tag, Stunde und Minute der Ausführung mit min/maxExclusive eingeschränkt. Hier muss aber min/maxInclusive angegeben sein.	implementiert	<p>(siehe auch Mail von Ralf Kunoth, "FinTS 4.0 Spezifikation - Anmerkungen Teil2", Punkt 1.)</p> <p>An einigen Stellen in den Schemas werden any-Elemente verwendet, die nicht in einem eigenen umschließenden Element, d.h ohne vorangehende Elemente, gekapselt sind. Bei der Schema-Validierung führt dieses zu Eindeutigkeitsproblemen, wenn ein auf der selben Syntaxebene vorhergehendes optionales oder wiederholbares Element vorliegt, welches im selben Namensraum wie das any-Element liegen kann. Die letzte XMLSpy-Version kann diese Uneindeutigkeit z.B. gar nicht mehr verarbeiten und erkennt es immer als Fehler. (Genaugenommen handelt es sich auch um einen Fehler, die meisten XML-Bibliotheken und Tools gehen damit allerdings recht tolerant um.)</p> <p>Das Problem kann behoben werden, indem das any-Element in einem eigenen einhüllenden Element gekapselt wird, oder indem ein entsprechendes Namespace-Constraint am any-Element für Eindeutigkeit sorgt. Die zweite Lösung hat den Vorteil, dass die eigentliche XML-Struktur nicht verändert wird, allerdings können im any-Element nicht mehr Elemente beliebiger Namespaces eingestellt werden.</p> <p>Die Einschränkung des Namespace-Constraints wäre die einfachste Möglichkeit das any-Element für die Geschäftsvorfallantwortdaten im komplexen Typ "Response" (in common.xsd) eindeutig validierbar zu machen. Die Angabe namespace=##other sorgt dafür, dass nur Geschäftsvorfallantwortdaten eingestellt werden können die nicht im types-Namespace definiert sind. Da das Element "RequestRespState" im types-Namespace definiert ist, wäre die Eindeutigkeit hergestellt.</p> <p>Als Nachteil hat dieses allerdings zur Folge, dass das Element "Response" in den spezialisierten Nachrichtentypen "SynchronisationResp" und</p>	Schema (Subscription-1.xsd): geändert

Nr.	Thema	Status	Bemerkungen	Wo?
			<p>"BankKeyResp" nicht mehr als Einschränkung des Basistyps "Response" definiert werden kann, da ja hier als Geschäftsvorfallantworten "SynchronisationResponse" bzw. "BankKeyResponse" eingestellt werden, die ja im types-Namensraum definiert sind.</p>	
14	<p>An einigen Stellen in den Schemas werden any-Elemente mit dem Namensraum-Constraint <code>##any</code> verwendet, die nicht in einem eigenen umschließenden Element, d.h. ohne vorangehende Elemente, gekapselt sind. Bei der Schema-Validierung führt dieses zu Eindeutigkeitsproblemen. Die letzte XMLSpy-Version kann es z.B. gar nicht verarbeiten und erkennt es als Fehler.</p>	implementiert	<p>(siehe auch Mail von Ralf Kunothe, "FinTS 4.0 Spezifikation - Anmerkungen Teil2", Punkt 2.)</p> <p>Gängige Schema-Validiertools lehnen die Ableitungs-Methodik für Auftragslisten als ungültig ab (Extension/Restriktion erfüllt nicht den Basistyp).</p> <p>Das Problem war bereits länger bekannt, wobei uns nicht klar ist, wodurch der Fehler entsteht. Das entsprechende Beispiel in der XMLSchema-Spezifikation, welches exakt diese Restriktions- oder Extensionstechnik beschreibt, wurde seinerzeit ebenfalls von nahezu allen gängigen Validiertools als ungültig angesehen. Im Prinzip rührt der Fehler daher, dass wir Basistypen durch mehrere Hierarchieebenen hindurch definieren (d.h. diese Basistypen enthalten ihrerseits wieder Basistypen) und in den nicht-abstrakten Typen und Elementen dann auch über mehrere Ebenen einschränken (d.h. Einschränkungen auf der äusseren Ebene und auch die enthaltenen Elemente, welche auf Basistypen beruhen, betreffend). Da meiner Meinung nach in den aktuellen Schemas alle Partikel der eingeschränkten Typen zum jeweiligen Partikel der Basistypen konform sind (d.h. jede Variante die für den eingeschränkten Typ Schema-valide ist, ist auch für den Basistyp Schema-valide), kann ich allerdings nicht erkennen, wo der Fehler im Detail liegt.</p> <p>Das Problem kann man leicht lösen, indem man auf die Definition der Basistypen verzichtet. Sie waren seinerzeit zur Verifizierung der "abgeleiteten" Typen gedacht. Ein weiterer Vorteil der Basistypen besteht darin, dass beim XML-Binding (z.B. in Java) Basistypen generiert werden, die für eine sinnvolle Architektur enorm wichtig sein können. (Beim XML-Binding kann ein Basistyp in eine abstrakte Basisklasse oder ein Interface umgesetzt werden, womit die Typsicherheit in Schnittstellen erhöht wird.)</p> <p>Im Schema message.xsd könnten somit alle abstrakten Basistypen gelöscht werden und die verwendenden Elemente können nicht mehr als Restriktion letzterer definiert werden. Die XML-Struktur des FinTS-Formats wird dadurch</p>	<p>Schemas (common.xsd, message.xsd): Namespace-Einschränkungen hinzugefügt</p>

Nr.	Thema	Status	Bemerkungen	Wo?
			<p>nicht verändert. Betroffen sind allerdings die Codegeneratoren von Bindingtools, welche aufgrund der Änderung andere Typhierarchien produzieren.</p>	
15	<p>Gängige Schema-Validiertools lehnen die Ableitungs-Methodik für Auftragslisten als ungültig ab (Extension/Restriktion erfüllt nicht den Basistyp).</p>	implementiert	<p>(siehe auch Mail von Ralf Kunoth, "FinTS 4.0 Spezifikation - Anmerkungen Teil2", Punkt 3.)</p> <p>Die Vorgabe war damals, die alten HBCI-MVE in allgemeinen Typen abzubilden und alle anderen Elemente lokal im jeweiligen GV zu definieren. XML-Bindingtools haben durch die lokal definierten Strukturen den Nachteil, dass man für in verschiedenen Geschäftsvorfällen gleichartig definierten Strukturen unterschiedliche Code-Klassen bekommt, d.h. man bräuchte zusätzliche Wrapper-Klassen, um diese Strukturen mit dem selben Code zu verarbeiten.</p> <p>Für eine sinnvolle Verwendung der Schemas mit XML-APIs zum Beispiel mit Xerces, benötigt man für die Geschäftsvorfall-Schemas in der Tat zusammenfassende Schemas, welche alle GV-Schemas eines Namespaces inkludieren (hier: alle Geschäftsvorfälle aus FinTS 4.0 im transaction-namespace). Das wurde schon mehrfach angemerkt, allerdings war die Meinung von Herrn Klement zu diesem Thema damals, das es sich um eine implementierungsspezifische Anforderung handelt und nicht in den offiziellen Schemas enthalten sein sollte.</p> <p>Bei der Codeerzeugung mit Bindingtools wird eine zusammenfassende Datei in der Regel nicht benötigt, da bei diesen Tools normalerweise nicht eine Datei pro Namespace angegeben wird (wie im XML-Schema import-Statement), sondern eine Liste aller Dateien, die in die Codeerzeugung eingehen sollen.</p> <p>Die beigelegten angepassten Schema-Dateien lassen sich z.B. problemlos mit XML-beans verarbeiten.</p>	<p>Schemas (common.xsd, structures.xsd, message.xsd): Basistypen gelöscht</p>
16	<p>Verschachtelung von Namensräumen und Fehler im XML-Bindingtools</p>	obsolet	<p>Betrifft die Modellierung der Auftrags- und Antwortlisten: In der Auftragsnachricht sind alle Auftragslisten-Elemente identisch mit dem XML-Tag "RequestList" benannt. Es existieren zwar globale Typdefinitionen für alle spezialisierten Auftragslisten ("StandardReqList", "BankKeyReqList", ..), allerdings ist es durch die Gleichnamigkeit der zugehörigen Elemente "RequestList" unmöglich, globale Elementedeklarationen vorzunehmen. Für die Validierung eines XML-Elements wird aber zwingend eine globale</p>	keine Änderung

Nr.	Thema	Status	Bemerkungen	Wo?
			<p>Elementdeklaration benötigt.</p> <p>In diesem Fall kann also nach der Entschlüsselung einer Auftragsliste nur der gesamte Nachrichtenkörper (mit der enthaltenen entschlüsselten Auftragsliste) validiert werden, oder aber man verwendet ein künstliches Hüllelement, welches man im Schema als globales Element deklariert und welches lediglich die zu validierende Auftragsliste enthält.</p> <p>Herr Klement hat auch dieses Problem seinerzeit als implementierungsspezifisch angesehen und von global deklarierten Auftragslisten abgeraten, da seiner Meinung nach nur solche Elemente global definiert sein sollten, die für sich gesehen (separat) transportiert werden dürfen.</p> <p>Die Validierung der entschlüsselten Antwortlisten stellt den Entwickler eines Clientsystems vor ähnliche Probleme, wobei allerdings für die Antwortlisten-Elemente nicht einmal globale Typdefinitionen existieren, welche direkt zur Deklaration von Hilfsstrukturen verwendet werden könnten.</p>	
17	Gleichnamige RequestList- und ResponseList-Elemente	obsolet	Es muss die Möglichkeit geben, Informationen zu mehreren TAN-Listen rückzumelden	keine Änderung
18	Wiederholbares Antwortsegment fehlt in TanListDisplay-1	implementiert		Schema (TanListDisplay-1.xsd): erweitert, [Syntax]