

# **Gutachten**

## **Die Anwendung des CDISC-ODM-Standards für die Archivierung klinischer Daten**

im Auftrag der  
**TMF - Technologie- und Methodenplattform für die vernetzte  
medizinische Forschung**



Version vom 10.02.2009  
(Übersetzung des englischen Originals)  
TMF-Produktnummer: P042031

Gutachter:

Dr. Jozef Aerts, XML4Pharma

© Lizenzbedingungen und Copyright für Gutachten und Berichte der TMF: Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Die Rechte liegen, sofern nicht anders angegeben, bei der TMF. Änderungen sind nicht zulässig. Eine Gewähr für die Richtigkeit der Inhalte kann die TMF nicht übernehmen. Eine Vervielfältigung und Weiterleitung ist ausschließlich innerhalb Ihrer Organisation oder Firma sowie der TMF-Mitgliedschaft erlaubt, sofern keine anders lautende Vereinbarung mit der TMF besteht. Aus Gründen der Qualitätssicherung und Transparenz bezüglich Verbreitung und Nutzung der TMF-Ergebnisse erfolgt die weitergehende Verbreitung ausschließlich über die TMF-Website oder die Geschäftsstelle der TMF.

TMF - Technologie- und Methodenplattform für die vernetzte medizinische Forschung e. V., Neustädtische Kirchstr. 6, 10117 Berlin / Tel. 030 31011950 / info@tmf-ev.de

## Inhaltsverzeichnis

1. Einleitung.....	3
1.1. CDISC ODM und XML.....	3
1.2. XML und Archivierung .....	3
2. CDISC-ODM-Konzepte für die Archivierung.....	6
2.1. Abschnitte in CDISC ODM-Dokumenten, die benutzt werden, um klinische Studien zu archivieren .....	6
2.2. Verknüpfen von Dateien.....	7
2.3. ODM und Audit-Trails.....	8
3. Vor- und Nachteile der Verwendung von ODM für die Archivierung .....	9
3.1. Vorteile der Verwendung von CDISC ODM für die Archivierung klinischer Studien....	9
3.2. Nachteile der Verwendung von CDISC ODM für die Archivierung klinischer Studien .	9
3.3. Verwendung von Text-, PDF- oder SAS-Transport-Dateien .....	10
3.4. Archivieren von SDTM-Datensätzen.....	11
4. Archivierungsstrategien .....	13
4.1. ODM 1.2 oder 1.3 für die Archivierung? .....	13
4.2. ODM-Archivierungsstrategien - übliche Praxis.....	14
4.3. Erfahrungen mit der Archivierung klinischer Studien .....	15
4.4. Archivierung von Datenbanken .....	15
4.5. CDISC ODM-Erweiterungen .....	16
4.6. Archivierung von Bildern und anderen elektronischen bzw. binären Daten .....	17
4.7. XML-Signatur .....	19
4.8. Elektronische Zertifikate mit XML-Signatur .....	20
4.9. Verwendung der XML-Signatur bei Archivierungsprojekten mit ODM .....	22
5. Nach dem Archivieren .....	23
5.1. Tools für das Arbeiten mit ODM-Archiven.....	23
5.2. Validierung von signierten ODM-Dateien.....	24
5.3. Archivierung und Gute Klinische Praxis (GCP).....	25
5.3.1. Beziehung zum Trial Master File (TMF).....	25
5.4. Fazit: Die langfristige Stabilität von CDISC-ODM für die Archivierung.....	25
6. Anhang.....	26
6.1. Literatur.....	26
6.2. Dateien der Begleit-CD zum Gutachten.....	27
6.2.1. XML-Signatur .....	27
6.2.2. CDISC ODM Vendor Extensions .....	27
6.2.3. Sample.....	27
6.2.4. Literature .....	27

## 1. Einleitung

Der CDISC-ODM-Standard wird heute allgemein als der weltweite offene Standard für den Austausch klinischer Daten und klinischer Metadaten anerkannt. Als solcher wird er häufig für die Erstellung portabler Studiendesigns [[Wheeldon, 2008](#)], für die automatische Einrichtung elektronischer Datenerfassungssysteme (EDC) [[Aerts, 2006](#)], und für den Austausch klinischer Daten und klinischer Metadaten verwendet.

Die aktuellste Version des CDISC-ODM-Standards ist Version 1.3, die Ende 2006 veröffentlicht wurde. Die populärste (am häufigsten implementierte) Version ist jedoch immer noch Version 1.2.

Es wird häufig behauptet, dass das CDISC-ODM-Format auch ideal für die Archivierung von vorhandenen klinischen Daten und Altdaten [[Cassels, 2001](#); [Kubick, 2005](#)] geeignet wäre. Es ist jedoch keine umfassende Analyse der Vor- und Nachteile der Verwendung des CDISC-ODM-Standards für die Archivierung klinischer Daten verfügbar. Das vorliegende Gutachten versucht genau diese umfassende Analyse zu erstellen.

### 1.1. CDISC ODM und XML

Der CDISC-ODM-Standard basiert auf XML und besitzt in seiner aktuellsten Version (1.3) ein zugehöriges XML-Schema. Version 1.2 verfügte über ein zugehöriges XML-Schema sowie eine Document Type Definition (DTD). XML-Schemas haben erhebliche Vorteile gegenüber DTDs. Es ist zum Beispiel nicht möglich, mit einer DTD komplexe Datentypen zu definieren. Version 1.1 und frühe Versionen von ODM verfügten nur über eine DTD. Das vorliegende Dokument basiert auf der Anwendung von ODM 1.2 und ODM 1.3. Frühere Versionen des CDISC-ODM-Standards werden nicht berücksichtigt.

Da der ODM-Standard viele Regeln enthält, die nicht in einem XML-Schema beschrieben werden können, besitzt jede Version des ODM-Standards ein zugehöriges Standard-Beschreibungsdokument, üblicherweise in Form einer HTML-Datei. Der Nachteil davon ist, dass dieses Standardbeschreibungsdokument nicht sehr gut maschinenlesbar ist, was bedeutet, dass zusätzliche Software geschrieben werden muss, um ODM-Dateien gegen den Standard<sup>1</sup> zu validieren.

Solch eine Software ist zurzeit von XML4Pharma gratis verfügbar, sie kann und sollte zur Gültigkeitskontrolle der erzeugten Archive im CDISC-ODM-Format verwendet werden.

### 1.2. XML und Archivierung

XML ist auf dem Weg dazu zum Standardformat für die Archivierung von Daten im Allgemeinen und wissenschaftlichen Daten im Besonderen [[Forrester, 2005](#)] zu werden. XML wurde auch als hervorragendes Format für die Archivierung klinischer Daten beschrieben

---

<sup>1</sup> Es wurde vom Autor vorgeschlagen, diese zusätzlichen Regeln mit Schematron zu beschreiben, einem relativ neuen XML-basierten Standard. Schematron basiert auf der Verwendung von XPath und Prüfpunkten. Diese Initiative ist jedoch noch sehr jung, und es konnten keine Ressourcen gefunden werden, um ein solches Projekt umzusetzen.

[Häber A., 2008]: "Eine Dokumentation und Rekonstruktion von Studienverlauf und Datenqualität sind möglich, vor allem, wenn die Studiensoftware den Datenexport als XML ermöglicht. Zusammengehörige Daten und Dokumente können in XML durch Verlinkung gemeinsam abgelegt werden. Zur Wahrung der Rechtssicherheit bietet XML die Möglichkeit der Archivierung von Stylesheets und digitalen Signaturen".

Konkrete Gründe für die Eignung von XML für die Archivierung sind:

XML ist ein offener, nicht proprietärer Standard, der durch das World Wide Web Consortium (W3C) entwickelt wurde. Dies bedeutet, dass es mehr oder weniger sicher ist, dass XML-Dateien auch in 50 Jahren noch von Computern gelesen und verstanden werden können.

XML besteht aus Daten und Metadaten, und die Metadaten (der Kontext der Daten) sind durch die Verwendung von Tags (Elementen) und Attributen direkt mit den Daten assoziiert.

Obwohl häufig erwähnt wird, dass der Vorteil von XML darin besteht, dass es auf ASCII basiert, entspricht dies nicht hundertprozentig der Wahrheit. Genauer wäre es zu sagen, dass XML auf Zeichen (im Gegensatz zu binär) basiert, die in der Vergangenheit üblicherweise als ISO-8859-1 Zeichen codiert wurden, aber inzwischen üblicherweise als UTF-8 (eine Unicode Kodierung) oder sogar als UTF-16 oder UTF-32 codiert werden. Die Verwendung von Zeichen bedeutet, dass XML grundsätzlich von jedem Computer lesbar ist, der Zeichen in einer der unterstützten Kodierungen lesen kann, was heute prinzipiell auf alle modernen Computer und Computersprachen zutrifft. Ein weiteres Ergebnis ist (bei Verwendung von Unicode), dass Daten in jeder beliebigen Schriftsprache archiviert werden können. Die Verwendung von Zeichen bedeutet auch, dass jedes XML-Dokument sowohl von Menschen gelesen werden kann als auch maschinenlesbar ist.

Um eine bessere Lesbarkeit zu erreichen, kann XML leicht in andere Formate transformiert werden, wie zum Beispiel HTML (Tabellen), Portable Document Format (PDF), Rich Text Format (RTF) oder sogar in MS Word Dokumente. Dies bedeutet auch, dass für jedes XML-Dokument eine Anzahl von Präsentationen erstellt werden können, abhängig davon, welche Informationen der Anwender sehen möchte und auf welche Art und Weise er sie angezeigt haben möchte. Die Sprache zur Beschreibung dieser Transformationen ist wiederum selbst ein XML-Standard, die so genannte „Extensible Stylesheet Transformation Language“ (XSLT).

Später werden wir auch sehen, dass XML nicht nur in Dateien, sondern auch in so genannten nativen XML-Datenbanken gespeichert werden kann. Dies bedeutet, dass für in XML erstellte Archive leicht Abfragen erstellt werden können.

Zusätzlich zu den bereits für XML genannten Vorteilen besitzt ODM mehrere spezifische Vorteile, die es besonders geeignet für die Archivierung klinischer Studien machen:

*es enthält die kompletten Daten der klinischen Studien,*

*es kann die kompletten Audit-Trail-Informationen enthalten,*

*es enthält die Metadaten der Prüfbögen (Case Report Form (CRF), aber auch die kompletten CRF-Daten jedes Studienteilnehmer (Probanden),*

*es ermöglicht, den Ablauf einer klinischen Studie nachzuvollziehen,*

*es wird von der FDA für die Einreichung von Probanden-CRFs akzeptiert (aktuell parallel zu PDF-Dateien).*

Üblicherweise werden die Daten klinischer Studien von einem klinischen Datenmanagementsystem (CDMS) oder einem Electronic Data Capture (EDC) System

exportiert. Immer mehr dieser Systeme ermöglichen einen direkten Export ins CDISC-ODM-Format, und einige davon sind bereits von CDISC dafür zertifiziert<sup>2</sup>. Aktuell implementieren auch Open Source EDC-Systeme wie OpenClinica den Datenexport im CDISC-ODM-Format. In einigen Fällen liegt jedoch die XML-Export-Datei in einem proprietären Format vor. In diesen Fällen kann die XML-Export-Datei häufig durch ein XSLT-Stylesheet und eine Transformation in das CDISC-ODM-Format konvertiert werden.

CDISC ODM ist jedoch mehr als nur ein einfaches XML-Format. Es kann auch als „Framework“ betrachtet werden, was bedeutet, dass man viele Dinge sehr flexibel in ODM beschreiben kann. Zum Beispiel kennt der CDISC-ODM-Standard das Konzept der „Forms“ (FormDef Element), schreibt aber nicht vor, welche „Forms“ (Formulare) verfügbar sind. Stattdessen erlaubt es durch die Verwendung von Attributen und Informationen in Unterelementen viele verschiedene Arten von „Forms“ mit verschiedenem Inhalt zu definieren. Dies ermöglicht die Wiederverwendung von „Visits“ (Untersuchungsterminen), „Forms“ (Formularen), Codelisten usw., was in einem kompakteren Studiendesign-Archiv resultiert.

---

<sup>2</sup> Eine Liste von CDISC-ODM-zertifizierten CDMS- und EDC-Systemen finden Sie unter <http://www.cdisc.org/odm/index.html>

## 2. CDISC-ODM-Konzepte für die Archivierung

### 2.1. Abschnitte in CDISC ODM-Dokumenten, die benutzt werden, um klinische Studien zu archivieren

Ab ODM Version 1.2 kann ein CDISC-ODM-Dokument (oder ein Dokumentensatz) grundsätzlich folgende Abschnitte enthalten:

*Study.* Dieser Abschnitt enthält den Aufbau der Studie in Form von Beschreibungen der Studientermine (StudyEvent), Formulare (Form), Unterformulare (ItemGroup) und Fragen (Item). Zusätzlich können grundlegende Definitionen wie Maßeinheiten, Code-Listen, Ableitungsverfahren<sup>3</sup> und Wertebereichsprüfungen definiert werden. Der „Studien“-Abschnitt ist durch die Verwendung von "MetaDataVersion"-Elementen versionierbar.

*AdminData* (Verwaltungsdaten). Dieser Abschnitt enthält Informationen zu den Nutzern (Prüfärzte, Monitore, usw.), deren Arbeitsplätzen und der Bedeutung ihrer Signatur (Unterschrift). Der Abschnitt AdminData ist nicht versionierbar.

*ReferenceData.* Dieser Abschnitt enthält alle klinischen Daten, die keinem einzelnen Probanden zugeordnet werden. Beispiel dafür sind Labor-Normalwerte usw. Die „ReferenceData“ Abschnitte besitzen Attribute, die sich auf die Version der verwendeten Metadaten beziehen.

*ClinicalData.* Dieser Abschnitt (gewöhnlich der umfangreichste) enthält die klinischen Daten. Diese werden in einer Baumstruktur pro Studienteilnehmer, pro Studientermin, pro Formular, pro Unterformular und pro Frage (Datenpunkt) aufgeführt, siehe Beispiel in Abb. 1. Das Element ClinicalData enthält Attribute, die sich auf die Version der verwendeten Metadaten beziehen.

*Association.* Dieser Abschnitt wurde in ODM 1.2 eingeführt und erlaubt es, zu geordneten Datenpaaren, die auf die eine oder andere Art miteinander verknüpft sind, eine Anmerkung hinzuzufügen. In der Praxis wird der Abschnitt Association allerdings selten verwendet.

*XML-Signature.* Ab ODM 1.2 ist es möglich, an das Ende der ODM-Datei eine elektronische Unterschrift anzufügen und damit den gesamten Inhalt der ODM-Datei als Ganzes zu signieren. Die Einführung dieser Funktion war keine Erfindung des ODM-Entwicklungsteams. Anstatt zu versuchen, „das Rad neu zu erfinden“, beschloss das Team, den existierenden XML-Signatur-Standard zu verwenden, der damals noch recht neu war, aber immer wichtiger wird. Die Verwendung von elektronischen Unterschriften für die XML-Archivierung im Allgemeinen und für CDISC ODM im Besonderen wird in [Abschnitt 4.9](#) beschrieben.

---

<sup>3</sup> In ODM 1.2 können Ableitungsverfahren mit „ImputationMethod“-Elementen beschrieben werden, Version 1.3 führte die viel leistungsfähigeren „MethodDef“-Elemente ein.

```

- <ClinicalData StudyOID="123-456-789" MetaDataVersionOID="v1.1.0">
+ <SubjectData SubjectKey="001">
- <SubjectData SubjectKey="002">
  <InvestigatorRef UserOID="USR.inv001" />
  <SiteRef LocationOID="LOC.site002" />
- <StudyEventData StudyEventOID="SE.VISIT0">
- <FormData FormOID="FORM.DEMOG">
  - <ItemGroupData ItemGroupOID="IG.DEMOG" ItemGroupRepeatKey="1" TransactionType="Insert">
    - <AuditRecord>
      <UserRef UserOID="USR.inv001" />
      <LocationRef LocationOID="LOC.site002" />
      <DateTimeStamp>2002-02-04T14:31:32-05:00</DateTimeStamp>
      <SourceID>46851404</SourceID>
    </AuditRecord>
    <ItemData ItemOID="IT.R_DRUG" Value="SDP" />
    <ItemData ItemOID="IT.TAREA" Value="ONC" />
    <ItemData ItemOID="IT.PNO" Value="143-02" />
    <ItemData ItemOID="IT.SCTRY" Value="USA" />
    <ItemData ItemOID="IT.F_STATUS" Value="S" />
    <ItemData ItemOID="IT.HT" Value="64" />
    <ItemData ItemOID="IT.WT" Value="153" />
    <ItemData ItemOID="IT.SEX" Value="F" />
    <ItemData ItemOID="IT.DOB" Value="1947-02-14" />
    <ItemData ItemOID="IT.RACE" Value="Black" />
    <ItemData ItemOID="IT.HTUNITS" Value="in" />
    <ItemData ItemOID="IT.WTUNITS" Value="lb" />
  </ItemGroupData>
</FormData>

```

Abb.1: Beispiel ODM

Jeder dieser Abschnitte kann in einem einzigen Dokument mehrfach vorkommen. Die Abschnitte können auch über mehrere Dateien (oder besser gesagt XML-Dokumente) verteilt sein. Dabei kann zum Beispiel eine Datei die Studien-Metadaten enthalten, eine andere die Verwaltungsdaten, eine die Referenzdaten (z. B. Normalwerte), eine die klinischen Daten des ersten Probandens, eine andere die klinischen Daten des zweiten Probandens usw.

Die Schlüssel, die die ODM-Metadaten, die Referenzdaten, die Verwaltungsdaten und die klinischen Daten miteinander verknüpfen, sind die so genannten OIDs (ODM IDs). Jedes ODM-Element, das etwas (in den Abschnitten „Study“ oder „AdminData“) definiert, besitzt eine eindeutige ID, die in seinem OID-Attribut enthalten ist. Es gibt sehr strenge Regeln für die OID-Eindeutigkeit, die im Standardbeschreibungsdokument gut beschrieben werden. Auf OIDs kann dann von Elementen in den Abschnitten ReferenceData und ClinicalData verwiesen werden. OIDs sind im Prinzip über Dateien hinweg gültig, weshalb ein Element in einer Datei auf eine OID (von der gleichen Studie) in einer anderen Datei verweisen kann.

## 2.2. Verknüpfen von Dateien

Wir werden später sehen, dass es oft vorteilhaft ist, klinische Studien nicht als ein einzelnes CDISC ODM-Dokument zu archivieren, sondern als eine Serie von ODM-Dokumenten. Die Informationen, die aufeinanderfolgende Dateien miteinander verbinden, sind in den Attributen „FileOID“ und „PriorFileOID“ enthalten. Das erste Attribut beschreibt die eindeutige ID der Datei in der aktuellen Studie, das zweite Attribut die „FileOID“ der in der Serie vorangegangenen Datei. Als solche ermöglichen „FileOID“ und „PriorFileOID“, die komplette Sequenz der Erstellung der ODM-Dateien<sup>4</sup> wieder herzustellen. Dies muss nicht

<sup>4</sup> Man könnte auch den Wert des „CreationDateTime“-Attributs benutzen, das im Wesentlichen ein „datetimestamp“ ist, aber dies ist aufwändiger und nicht immer ausreichend.

immer eine lineare Sequenz sein, es kann auch eine Verzweigungssequenz abgebildet werden, d. h. zwei Dateien können die gleiche „PriorFileOID“ besitzen, wenn sie zwei Zweige repräsentieren, die den gleichen Verzweigungspunkt wie die vorangegangene Datei besitzen. Die Verknüpfung von Dateien macht einen wichtigen Teil der Strategie bei der Archivierung von klinischen Studien im CDISC ODM-Format aus, weil eine einzelne große Datei schwieriger zu handhaben ist, als mehrere kleinere, aber verknüpfte Dateien.

### 2.3. ODM und Audit-Trails

Eine der wichtigsten „Anwenderanforderungen“ an die erste Version der ODM war, dass es komplette Audit-Trails erlauben und 21 CFR Part 11 Compliance ermöglichen sollte [Aerts, 2006]. Deshalb wurde ein Mechanismus entwickelt, der Audit Trails auf allen Ebenen der klinischen Datenhierarchie ermöglicht. Abb. 2 zeigt, wie dies implementiert wurde: Eine oder mehrere AuditRecord-Elemente können auf den Ebenen der SubjectData (alle Daten eines einzelnen Probanden), StudyEventData (alle Daten eines einzelnen Studientermins für einen bestimmten Probanden), FormData, ItemGroupData und ItemData hinzugefügt werden. Dies bedeutet, dass man einen Audit Trail nicht nur zu einem einzelnen Datenpunkt oder einem einzelnen Formular, sondern auch zu allen Daten eines einzelnen Probanden hinzufügen kann.

Das Element AuditRecord ermöglicht es, alle notwendigen Informationen über die Änderung der Daten hinzuzufügen: wer, wo, wann (Datums-Zeitstempel) und warum. Die dafür verantwortlichen Unterelemente sind UserRef, LocationRef, DateTimeStamp, ReasonForChange und SourceID. Eine Beispieldatei mit einigen Audit-Records (Audit Datensätzen) finden sich im Anhang (Datei zbi.net-2004-02-27-oncology002.xml).

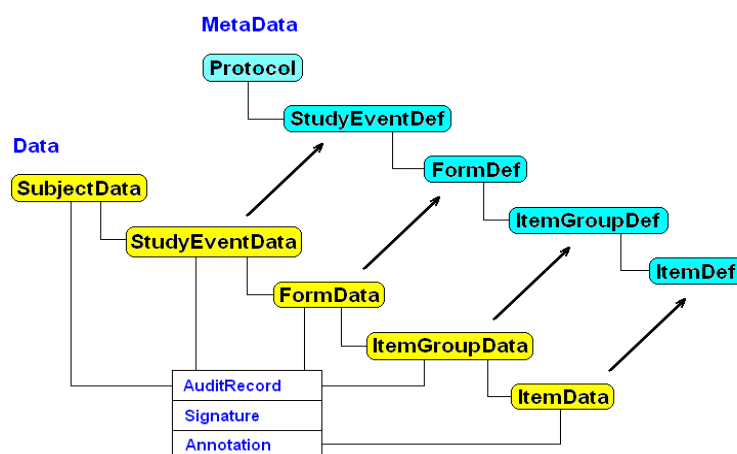


Abb. 2: Audit Trail in ODM



### 3. Vor- und Nachteile der Verwendung von ODM für die Archivierung

#### 3.1. Vorteile der Verwendung von CDISC ODM für die Archivierung klinischer Studien

Die größten Vorteile der Verwendung von CDISC ODM für die Archivierung sind:

- CDISC ODM ist ein offener, international anerkannter Standard,
- er wird von der FDA akzeptiert,
- er basiert auf XML, so dass Daten und Metadaten in der gleichen Struktur und Datei gespeichert werden können, und
- er ermöglicht die Speicherung kompletter Audit Trails

#### 3.2. Nachteile der Verwendung von CDISC ODM für die Archivierung klinischer Studien

Wie jedes XML-Dokument können CDISC-ODM-Dateien manchmal recht groß werden. Der Grund für diese Eigenschaft liegt darin, dass die Dateien sowohl die Daten als auch die Metadaten enthalten. Deshalb ist eine zweckmäßige Archivierungsstrategie erforderlich. Außerdem kann besonders bei Legacy-Systemen das Mappen auf und die Erstellung von ODM-XML ziemlich kompliziert werden.

Nach unseren Erfahrungen erfordert das Mappen von einem Legacy-System auf ODM ungefähr einen Mitarbeitermonat an Arbeit, wobei die erforderliche Programmierung (Schreiben der SQL-Queries, die Entwicklung der XSLT-Transformation Stylesheets, usw.) nicht inbegriffen ist.

Ein anderer Nachteil besteht darin, dass die Darstellung der CRFs verloren geht. Dies kann (und wird üblicherweise) dadurch gelöst, dass leere CRFs als PDF zusammen mit dem ODM gespeichert werden. In der ODM-Datei kann durch Verwendung des Elements „ArchiveLayout“ für jede CRF (und jede Sprachversion) eine Verknüpfung zum PDF wie folgt hinzugefügt werden:

```
<FormDef OID="FM.VITALSIGNS" Name="Vital Sign Form">
  <ItemGroupRef ... />
  <ArchiveLayout OID="FM.VITALSIGNS.LAYOUT.EN"
    PdfFilename="Form_Vitals_blank_english.pdf">
  </ArchiveLayout>
  <ArchiveLayout OID="FM.VITALSIGNS.LAYOUT.EN"
    PdfFilename="Form_Vitals_blank_german.pdf">
  </ArchiveLayout>
</FormDef>
```

Ein Beispiel, das einen Link auf eine englische und deutsche PDF-Datei enthält und die Darstellung des Formulars verdeutlicht.

Falls erforderlich, kann auf das verwendete Formular-Layout dann im klinischen Datenabschnitt verwiesen werden:

```
<FormData FormOID="FM.VITALSIGNS" ...>
  <ArchiveLayoutRef ArchiveLayoutOID="FM.VITALSIGNS.LAYOUT.EN"/>
  <ItemGroupData> ... </ItemGroupData>
</FormData>
```

### 3.3. Verwendung von Text-, PDF- oder SAS-Transport-Dateien

Für die Archivierung klinischer Studien kommen nur drei echte Alternativen zum XML-Format in Betracht.

Erstens kann man klinische Studien natürlich in einfachen Textdateien (wie Komma- oder Doppelpunkt-separierte Textdateien - CSV) archivieren. Hier besteht natürlich das Problem der Metadaten. Es ist sehr schwierig, wenn nicht sogar unmöglich, eine Struktur zu schaffen, die die Daten und auch die Metadaten beschreibt, ohne XML zu verwenden.

Eine zweite Alternative besteht darin, die klinischen Daten als PDF-Datei zu archivieren. Zum Beispiel könnte für jedes ausgefüllte Formular eine PDF-Datei erstellt werden, zu der zusätzliche Informationen hinzugefügt werden, wie zum Beispiel der Studientermin, an dem das Formular verwendet wurde, das Datum und die Zeit des Studientermins und Informationen über den Prüfarzt. Dann könnten separate PDF-Dateien mit allen Metadaten und Verwaltungsdaten erstellt werden.

Hinterher treten dadurch jedoch mehrere Probleme auf:

- es gibt keine standardisierte Archivierung - jedes Unternehmen würde sein eigenes Verfahren verwenden, wodurch es mit der Zeit sogar zu Unterschieden innerhalb eines Unternehmens kommen könnte.
- es ist schwierig, in PDF-Archiven Queries durchzuführen.
- es besteht die Gefahr, dass die Ergebnisse der Queries nicht enthalten sind und komplette Audit Trails über mehrere Dokumente verteilt werden.

Die Archivierung als PDF-Dateien ist meiner Meinung nach ein hervorragendes Verfahren, um klinische Studien zu archivieren, bei denen nur Papier zum Einsatz kam. Aber selbst dann hat dies auf eine ordnungsgemäße Art und Weise zu erfolgen. Wir wissen von einem Unternehmen, das seine alten klinischen Studien durch Scannen der Papier-CRFs archivierte, wobei es die Scans als TIFF-Dateien<sup>5</sup> speicherte und diese dann in PDF-Dateien umwandelte. Das Ergebnis war, dass die Texte in den PDF-Dateien nicht mehr durchsuchbar waren, da sie nur ... Bilder enthielten.

SAS Transport 5 ist mehr oder weniger ein de facto Standard für klinische Daten in der Pharmaindustrie. Der Grund dafür ist, dass es das bevorzugte Format der FDA für die Einreichung von Daten ist. SAS Transport 5 war das einzige SAS-Format, das jemals durch das Unternehmen SAS „offen“ gelegt wurde (d. h. die Details dieses binären Formats wurden von SAS als ein kurzes Dokument veröffentlicht [[SAS, TS-140](#)]). Da SAS Marktführer im Bereich „statistische Analyse“ ist und jede SAS-Version außer den ältesten SAS Transport-

---

<sup>5</sup> Im Gesundheitswesen ist TIFF immer noch ein verbreitetes Format zur Archivierung von Krankenakten. Es gibt allerdings inzwischen Software (z. B. von eDocfile, Omniformat, Aquaforest, ADLibsoftware - und viele andere), die eine Batch-Umwandlung von TIFF-Dateien in durchsuchbare PDFs oder einfache Textdateien ermöglicht.

Dateien lesen und schreiben kann, wurde dieses Format sehr verbreitet für den Austausch klinischer Daten<sup>6</sup>.

SAS Transport 5 hat jedoch viele Nachteile:

- Es ist ein binäres Format, wodurch es von Menschen nicht lesbar ist.
- Das Zahlenformat ist ein Format, das aus der Zeit der IBM Großrechner stammt und heute von keiner modernen Programmiersprache mehr unterstützt wird.
- SAS Transport 5 kennt nur zwei Datentypen: Text und numerisch.
- Variablen-Namen sind auf 8 Zeichen, Labels auf 40 Zeichen begrenzt.
- Variablen-Werte sind auf 200 Zeichen begrenzt, was bedeutet, dass Texte, die länger als 200 Zeichen sind, über mehrere Variablen oder sogar über verschiedene Dateien verteilt werden müssen.
- Nur eine begrenzte Zahl an Metadaten kann in SAS Transport-Dateien (im Header) eingefügt werden. Alle anderen Metadaten müssen in getrennte Dateien aufgenommen werden.
- SAS Transport 5-Dateien können nur mit (teurer) SAS Software effektiv verwaltet oder durchsucht werden.
- Es gibt keine standardisierte Methode zur Archivierung mit SAS; jedes Unternehmen wird dazu seine eigenen Verfahren verwenden.
- Es können keine Audit Trails direkt auf Datenpunkt-Ebene gespeichert werden.

Man kann argumentieren, dass SAS Transport-Dateien im Vergleich zu XML wesentlich kompakter sind. Nach unseren Erfahrungen ist dies nicht der Fall. Auf Grund seiner internen Struktur „verschwendet“ SAS Transport viel Platz auf dem Speichermedium, weshalb die Datei ungefähr genau so groß ist wie die, die das XML-Format für die Darstellung derselben Daten benötigt.

### 3.4. Archivieren von SDTM-Datensätzen

*Man könnte sich auch fragen, warum klinische Datenbanken und Datenverwaltungssysteme oft archiviert werden und SDTM-Datensätze oder -Datenbanken nicht. Im Wesentlichen entspricht diese Aussage nicht der Wahrheit.* Unternehmen archivieren auch SDTM-Datensätze, da aber SDTM ein relativer junger Content-Standard ist, gibt es nicht viele veröffentlichte Informationen über die Archivierung unter Verwendung von SDTM.

Da SAS Transport das von der FDA bevorzugte Format für die Einreichung von Daten ist, liegen die SDTM-Informationen im SAS Transport 5 Format vor. Die Metadaten sind allerdings schon im XML-Format, d. h. die Datei define.xml enthält die Metadaten der Einreichung in SDTM-SAS. Define.xml (der offizielle Name des Standards ist CRT-DDS: Case Report Tabulation Data Definition Specification) basiert auf ODM: Nach dem ODM Vendor Extension Mechanismus handelt es sich dabei um eine ODM Erweiterung (siehe weiter).

CDISC hat schon ein XML-ODM-basiertes Format zum Speichern der Einreichungsdaten entwickelt, das kompatibel mit define.xml ist. Die FDA hat jedoch dieses Format abgelehnt und ein Projekt gestartet, um eine HL7-XML-Message für Einreichungsdaten zu entwickeln<sup>7</sup>.

---

<sup>6</sup> Auch andere Statistikpakete, wie zum Beispiel das Open Source Paket R-Software, können SAS Transport 5 lesen und schreiben.

<sup>7</sup> Persönlich glaube ich, dass dies eine sehr schlechte Idee ist. Man braucht nicht nur Jahre, um HL7-XML-Messages zu entwickeln, sie sind gewöhnlich auch übermäßig kompliziert und erfüllen die ISO-Standards nicht. In der Praxis werden HL7-XML-Messages kaum benutzt, und es wird äußerst schwierig sein, eine HL7-XML-Message zu entwickeln, die auch mit define.xml vereinbar ist.

Als solches wird die Archivierung der klinischen Daten zusammen mit den Einreichungsdaten unter einem Standard (noch) nicht oft durchgeführt, da beide verschiedene Formate verwenden. Dies wird auch nicht häufiger werden, solange SAS Transport 5 das bevorzugte Format der FDA ist, und solange sich die FDA weigert, eine ODM-Erweiterung (wie sie bereits durch CDISC entwickelt wurde) zur Speicherung der Einreichungsdaten zu akzeptieren.

Einige Unternehmen transformieren bereits SDTM SAS Transport-Dateien in XML, aber hauptsächlich um damit ein SDTM-Datawarehouse zu bevölkern. Das XML-Format ist gewöhnlich nicht ODM, sondern ein XML-Format, das die Datenbankstruktur der SDTM-Datenbank oder des Data Warehouses darstellt.

## 4. Archivierungsstrategien

### 4.1. ODM 1.2 oder 1.3 für die Archivierung?

Für die Archivierung der klinischen Daten und der Metadaten steht einem prinzipiell die Wahl zwischen Version 1.2 oder 1.3 des CDISC-ODM-Standards offen. Während ODM 1.2 allgemein akzeptiert ist und eine ganze Menge an Tools dafür zur Verfügung steht, verfügt ODM 1.3 über einige besondere Funktionen, die die Archivierung erleichtern<sup>8</sup>. Diese neuen Funktionen sind:

- Typisierte Datenübertragung: Während ODM 1.2 keine Informationen über den Datentyp eines Datenpunktes im Abschnitt ClinicalData enthält (weshalb für eine Validierung des Datenpunktes ein Lookup der Metadaten erforderlich ist), erlaubt ODM 1.3 eine Übertragung typisierter Daten. Dies kann am besten in Form eines einfachen Beispiels für einen Datenpunkt vom Typ Integer erklärt werden:

ODM 1.2: `<ItemData ItemOID="IT.AGE" Value="67"/>`

ODM 1.3: `<ItemDataInteger ItemOID="IT.AGE">67</ItemDataInteger>`

Bitte nehmen Sie zur Kenntnis, dass der ODM 1.2 Mechanismus auch in ODM 1.3 gültig ist; es ist jedoch nicht erlaubt, typisierte und untypisierte Datenpunkte in einer einzelnen ItemDataGroup zu mischen.

Der Vorteil des ODM 1.3 Mechanismus bei der Archivierung besteht darin, dass die Daten gegen den Datentyp validiert werden können, ohne dass ein Lookup in den Metadaten (die sich in einer anderen Datei befinden können) erforderlich ist. Dadurch lässt sich schon beim Überprüfen der Daten im Archiv ihr Datentyp ablesen.

- Gruppieren von Audit Records, Anmerkungen und Signaturen. Während in ODM 1.2 jeder Audit-Record (Element AuditRecord), jede Anmerkung (Element Annotation) oder Signatur (Element Signature<sup>9</sup>) zum Datenpunkt selbst (oder der Gruppe der Datenpunkte) hinzugefügt werden muss, ermöglicht ODM 1.3 es, alle Audit Records, Anmerkungen und Signaturen in einem oder mehreren AuditRecords, Annotation- bzw. Signatur-Elementen zu gruppieren, die allen SubjectData-Elementen im ClinicalData-Element folgen. Auf dem Niveau des Datenpunktes wird ein einfacher Verweis auf eine ID (IDREF) eingegeben<sup>10</sup>. Zum Beispiel:

`<ItemDataIntervalDatetime ItemOID="ID.IDT" AuditRecordID="Audit-`

<sup>8</sup> Die wichtigsten Verbesserungen in ODM 1.3 finden sich im Bereich Internationalisierung und EDC, [Aerts, 2006], die hier nicht weiter diskutiert werden.

<sup>9</sup> Nicht zu verwechseln mit einer elektronischen Unterschrift. In ODM ist das einfache Signatur-Element nur eine Annotation, dass der Anwender die Daten auf irgendeine Art und Weise signiert hat.

<sup>10</sup> In der XML-Schema-Sprache ist das Attribut AuditRecordID eine IDREF, und das Attribut ID im AuditRecord ist eine ID, was bedeutet, dass ein Schema Fehler gemeldet wird, wenn eine IDREF gefunden wird, für die es keine ID gibt.

```
001">1959-12-11/2003-11-07T16:24</ItemDataIntervalDatetime>

<AuditRecords>
  <AuditRecord ID="Audit-001">
    <UserRef UserOID="USR.inv001" />
    <LocationRef LocationOID="LOC.site001" />
    <DateTimeStamp>2002-02-04T14:11:32-05:00</DateTimeStamp>
    <SourceID>46881304</SourceID>
  </AuditRecord>
</AuditRecords>
```

Abhängig von der Archivierungsstrategie, der Art und Weise, wie Audit Records in der Datenbank organisiert werden, und den verfügbaren Software Tools kann man sich entscheiden, ob man den ODM Standard in Version 1.2 oder Version 1.3 verwenden möchte. Man kann sich auch entscheiden, für die Archivierung ODM 1.3 zu verwenden, aber nicht alle neuen Funktionen von ODM 1.3 zu implementieren (da es eine Abwärts-Kompatibilität gibt). Wenn sich zum Beispiel alle Audit Records in der Datenbank in einer einzelnen Tabelle befinden, kann die Archivierung mit der neuen Gruppierungsfunktion vorteilhaft sein (Leistungsverbesserung).

#### 4.2. ODM-Archivierungsstrategien - übliche Praxis

Die am häufigsten verwendete Verfahrensweise in der Pharmabranche bei der Archivierung klinischer Studien ist die Archivierung: Pro Studie, pro Probanden. Dies bedeutet, dass für jede Studie ein Archiv erstellt wird und dass innerhalb dieser Studie eine ODM-Datei für jeden Probanden erstellt wird. Der Grund dafür, dass nicht eine Datei für die gesamte Studie angelegt wird, ist ein sehr praktischer: die Dateigröße. Obwohl Festplattenspeicher heutzutage üblicherweise kein größeres Problem mehr darstellt, ist es die Größe des verfügbaren Arbeitsspeichers beim Anzeigen und Durchsehen der Dateiinhalte immer noch. Wie wir später beschreiben werden, benötigt der Aufbau der XML-Baumstruktur im Arbeitsspeicher ungefähr die 10-20fache Menge an Speicher, im Vergleich zur Größe der Datei.

Dies ist auch der Fall beim Signieren einer XML-Datei mittels XML-Signature (siehe weiter unten).

Die häufigste Vorgehensweise besteht darin, die Metadaten zuerst zu archivieren. Falls verschiedene Versionen der Metadaten (MetaDataVersion OID) existieren, werden diese entweder alle in einer einzelnen Datei zusammengefasst, oder eine einzelne Version der Metadaten wird in einer einzelnen Datei gespeichert. Wenn Basisdesign und die Aktualisierungen des Studiendesigns archiviert werden, sollte man nicht vergessen, die „Include“-Statements mit aufzunehmen, die anzeigen, dass ein Studiendesign ein älteres Studiendesign enthält.

Falls es jedoch nur ein einziges Studiendesign für die Studie gibt (auch wenn es aktualisiert wurde), entspricht es der guten Praxis, (zusätzlich) eine einzelne, umfassend beschreibende MetaDataVersion zu haben, d. h. eine Version, bei der das Include Statement ausgeführt und die vorherigen Versionen mit eingeschlossen wurden<sup>11</sup>.

---

<sup>11</sup> In einigen Studien kann es verschiedene Gruppen geben, jede mit ihrer eigenen Version der Metadaten.

Es ist wichtig, eine Strategie zu haben, die Datei-Sequenzen erzeugt. Denken Sie daran, dass Sie dafür die Attribute FileOID und PriorFileOID des höchsten ODM-Elements verwenden müssen.

Zweitens werden die Verwaltungsdaten archiviert (AdminData). Da diese nicht versioniert sind, werden sie üblicherweise in eine einzelne ODM-Datei gespeichert.

Drittens werden die Referenzdaten (ReferenceData) archiviert. Bitte beachten Sie, dass es mehrere Versionen (MetaDataVersionOID) der Referenzdaten geben kann.

Zuletzt (falls kein Archiv mit Verknüpfungen (Association-Element) erstellt wird, was gewöhnlich der Fall ist), werden die klinischen Daten archiviert, dabei werden üblicherweise jeweils die Daten eines Probandens in einer Datei gespeichert. Manchmal enthält jede Datei mit klinischen Daten auch den entsprechenden MetaDataVersion(s)-Abschnitt, um die Visualisierung und Validierung zu vereinfachen (damit keine Suche der Metadaten in anderen Dateien erforderlich wird).

#### **4.3. Erfahrungen mit der Archivierung klinischer Studien**

Obwohl wir mehrere Archivierungsprojekte kennen, in denen klinische Studien in CDISC ODM archiviert wurden, gibt es nur einen gut dokumentierten Fall, von dem dies bekannt ist. Auf dem European CDISC Interchange, der 2005 in Paris stattfand, hat Wayne Kubick von Lincoln Technologies (jetzt PhaseForward) über die Ergebnisse eines Archivierungsprojekts bei Abbott Laboratories berichtet. Bei dem Projekt wurden mehr als 3500 Studien-Datenbankarchive<sup>12</sup> als CDISC ODM archiviert (eine Kopie der Präsentation finden Sie im Anhang). Die Daten bezogen sich auf einen Zeitraum von 20 Jahren und das Projekt wurde innerhalb von ungefähr 13 Monaten abgeschlossen. Die eigentliche Archivierung wurde von einem Team von fünf Mitarbeitern in ungefähr acht Monaten durchgeführt.

Die Hauptprobleme, die bei diesem Projekt auftraten, betrafen nicht die tatsächliche Transformation nach XML, sondern hatten mehr damit zu tun, die Metadaten für die Daten zu identifizieren. Ein anderes Problem bestand darin, dass es manchmal sehr schwierig war, die verschiedenen Audit Trail-Quellen zusammenzuführen. Auch die Vielfalt der Datenformate und Datenstrukturen, die in diesen 20 Jahren verwendet wurden, war nicht immer einfach zu handhaben.

Aus dem persönlichen Schriftverkehr mit einigen Personen, die an diesem Projekt teilgenommen haben, wissen wir, dass die Dateigröße am Anfang ein Problem darstellte, das durch die Erstellung einer ODM-Datei pro Probanden und pro Studie gelöst wurde.

#### **4.4. Archivierung von Datenbanken**

In diesem Zusammenhang muss ein wichtiges Problem angesprochen werden. In den meisten Fällen werden die klinischen Daten die archiviert werden sollen, in klinischen

---

<sup>12</sup> Aus der Präsentation geht nicht eindeutig hervor, was unter einem „Studien-Datenbankarchiv“ zu verstehen ist.



Datenbanken (üblicherweise relationalen Datenbanken<sup>13</sup>) gehalten. Um diese Daten in das XML-basierte CDISC ODM zu konvertieren, können mehrere Strategien verfolgt werden.

Eine einfache, aber effektive Strategie besteht darin, den kompletten Inhalt der Datenbanktabellen in eine oder mehrere XML-Dateien zu laden. Die meisten modernen SQL-Dialekte verfügen dafür über ein spezielles Schlagwort. Zum Beispiel bei MS-SQLServer:

```
SELECT * FROM TABLENAME FOR XML [keywords]
```

Oracle hat seine „XML SQL Utility“ (als XSU abgekürzt), aber man kann auch SQL/XML Funktionen [<http://www.oracle.com/technology/oramag/oracle/03-may/o33xml.html>], wie XMLForest und XMLElement für die Erstellung komplexer Queries benutzen, um komplexe XML-Strukturen zu erzeugen.

Nachdem die Tabellen als XML ausgegeben wurden, kann XSLT verwendet werden, um die Datenbank-XML nach CDISC ODM zu transformieren. Zur Erzeugung der XSLT muss man zuerst das, was ich ein Mapping-Dokument bezeichne, erstellen, d. h. eine Beschreibung, wie jedes Feld in der Datenbank auf den CDISC-ODM-Standard abgebildet wird. Dies ist üblicherweise der schwierige Teil der Arbeit. Nach meiner eigenen Erfahrung erfordert die Erstellung eines solchen Mapping-Dokuments wenigstens 2 bis 3-Mitarbeiter-Wochen, sogar für einen Spezialisten. Daher wird üblicherweise empfohlen, dass diese Arbeit von einem Team aus zwei Personen durchgeführt wird, eine, die die Datenbankstruktur kennt und sehr gut versteht, und eine, die die ODM-Struktur und den ODM-Standard sehr gut kennt. Durch Verwendung des Mapping-Dokuments kann dann die XSLT entwickelt werden, die erforderlich ist, um die Datenbank-XML nach CDISC ODM zu transformieren. Folglich ist es im Prinzip möglich, Datenbankinhalte nach ODM zu konvertieren.

Es kann jedoch auch vorkommen, dass bestimmte Aspekte einer klinischen Studie oder einer klinischen Datenbank nicht gut auf den ODM-Standard gemappt werden können. Zum Beispiel kann ein Formular Attribute besitzen, die in ODM<sup>14</sup> nicht erfasst werden. Wenn solche zusätzlichen Eigenschaften existieren, kann man daran denken, eine Erweiterung des ODM-Standards zu schreiben.

#### 4.5. CDISC ODM-Erweiterungen

Auf den ersten Blick könnte man argumentieren, dass die Verwendung einer ODM-Erweiterung eine schlechte Idee ist, da man damit prinzipiell vom ODM-Standard abweicht. Das ist so nicht richtig: Die Verwendung einer ODM-Erweiterung ändert nichts am Standard, sondern man erweitert ihn damit.

Die Archivierung mit einer ODM-Erweiterung ist eine ausgezeichnete Lösung, solange die Erweiterung gut dokumentiert ist. Dafür wurde vom ODM-Team der ODM-Schema-Erweiterungsmechanismus entwickelt<sup>15</sup>. Er beruht auf folgenden Prinzipien:

---

<sup>13</sup> Einige der neueren EDC-Systeme benutzen schon native XML-Datenbanken, um die Daten als XML zu speichern. Ein Beispiel ist Nextrials, dessen EDC-System PRISM die native XML-Datenbank Tigerlogic von RainingData benutzt.

<sup>14</sup> Man muss berücksichtigen, dass ODM ein „Konsens“-Standard ist, mit dem versucht wird, die Mindestzahl von Funktionen zu bieten, auf die sich alle Parteien einigen können.

<sup>15</sup> Ab ODM 1.2 und unter Verwendung des ODM 1.2.1 XML-Schemas. Für ODM 1.3 wurde das XML-Schema bereits dem gleichen Mechanismus entsprechend als erweiterbar entwickelt.



- Alle neuen Elemente und Attribute existieren in einem separaten Namensraum.  
Die neuen Elemente und die Attribute werden in einer separaten XML-Schema-Datei beschrieben. Bevorzugt wird ein Dateiname, der auf „-ns.xsd“ endet, benutzt, zum Beispiel „myextension-ns.xsd“.
- Die Art und Weise, wie die neuen Elemente und die Attribute mit den bestehenden ODM-Elementen verbunden sind, wird in einer separaten XML-Schema-Datei beschrieben. Bevorzugt wird ein Dateiname, der auf „-extension.xsd“ endet, benutzt, zum Beispiel „myextension-extension.xsd“.
- Alle Schemas und Namensräume werden über eine sehr kleine Schema-Datei zusammengehalten, die nur die Namensräume definiert und über ein XInclude Statement auf die Datei „myextension-extension.xsd“ verweist.  
Der Name der Erweiterung wird bevorzugt als Dateiname (z. B. „myextension.xsd“) verwendet.  
Diese Datei ist es, auf die verwiesen werden muss, wenn eine XML-Schema-Validierung der XML Instanz-Datei(en) mit klinischen Daten durchgeführt wird.

Der Anhang enthält ein sehr einfaches Beispiel für eine solche Erweiterung (Verzeichnis „CDISC\_ODM\_1\_3\_Vendor\_Extensions“). Sie enthält die folgenden Dateien:

- my-simple.xsd : die „Kleber“ XML-Schema-Datei - Referenzdatei für die Validierung gegen das Schema
- my-simple-ns.xsd : die XML-Schema-Datei mit den neuen Elementen und Attributen
- my-simple-extension.xsd : die Schema-Datei, die beschreibt, wie die neuen Elemente und die Attribute mit den ODM Elementen verknüpft sind
- my-simple.xml : ein Beispiel ODM Instanz-Datei mit Erweiterungen
- alle anderen erforderlichen ODM 1.3 Dateien

Es ist sehr wichtig, das XML-Schema mit der Erweiterung gemäß den oben beschriebenen Prinzipien und dem dort beschriebenen Mechanismus zu entwickeln. Dies ermöglicht nicht nur eine Validierung der erstellten XML-Datei mit klinischen Daten gegen das Schema, sondern sie ermöglicht es, später auch die „Regeln“ der Erweiterung durchzusehen und nachzuvollziehen. Selbstverständlich sind dazu zusätzliche Informationen über das Warum und Wie der Erweiterung erforderlich.

#### **4.6. Archivierung von Bildern und anderen elektronischen bzw. binären Daten**

Es kommt immer häufiger vor, dass in klinischen Studien nicht nur einfache Text- oder numerische Daten („Datenpunkte“) erfasst werden, sondern auch Bilder (zum Beispiel Röntgen- oder CT-Aufnahmen) und andere elektronische bzw. binäre Daten (z. B. EKGs, Spirometriedaten). Die Frage ist, wie können diese mit Hilfe von CDISC ODM zusammen mit den „einfachen“ Daten archiviert werden?

Es gibt zwei Strategien, die verfolgt werden können.

Die erste Strategie besteht darin, die binären Daten in die ODM-Dateien selbst aufzunehmen. Ab Version 1.3 verfügt das CDISC ODM über folgende binäre Datentypen: HexBinary, base64Binary, hexFloat und base64Float. Besonders der Datentyp base64Binary kann

benutzt werden, um binäre Daten in einer ODM-Datei zu speichern. Dafür werden die binären Daten (z. B. ein JPEG-Bild) in ein base64-Zeichenformat transformiert, das in den Inhalten des Elements „ItemDataBase64Binary“ verwendet wird.

Zum Beispiel kann das CDISC Logo mit einer base64-Kodierung folgendermaßen als Datenpunkt dargestellt werden:

```
<ItemDataBase64Binary
ItemOID="IT.CDISCLOGO">data:image/gif;base64,ROIgODIhpQA5APcAAAAAP/////+/7//3S
mmhdBaf/7/v39/g01ZBE6bBc6ZAKyXggpTw03aBA6aRM/cRI4YhdCcRdBbhU6YhIAbBo/aiA8XA
IrVQYxXAK3ZQgwWAO2YgozWw0/cA8/cg04Yg02YA45ZBA/bg81XQ4xVhNCchI/bRE6ZRE6YxQ
+ahVBbRQ+ZxI3XREzVhdBahhCaxU6XhhBaRtAZxg5WxtAZRs+YR1BZRs8Xh5BZB9DZyNEZSh
McChIaCtLbC1KZzRScDdWdUffVx0jL/N297k6uru8gU0YQg5aAo8aQk1Xgo5ZQ08aQ48ZhA+a
Q85XxFAahJBbRE9Zg82WhRBaRM8YxVBahZCbDcaRdBaRQ4WRY9YRhBZhpFbRhAZB5GaiFG
aSVJayxSdDBPbTtad0tqh1RthWR8kmuCI3OInIscsnyQo4aZqp2suqSyvwo5YhNEbxVCaRZBZBI
FaxhCZ1Z2kLvGzxNEaRhDZRpeZRxGZh9KbB1FY42hsJWntau5xMXP18zU2g03VBVAXUZnfrTBy
sDL08jT29DZ387X3cnS2OHn6xhEYNTc4R5KYyRPaCpVbcrU2ufs79vj5...
</ItemDataBase64Binary>
```

Man sieht sofort den Nachteil: Die Informationen innerhalb des ODM-Elements werden als Textinhalt sehr groß. Das CDISC-Logo benötigt, wenn es in base64 transformiert wird, beinahe 8000 Zeichen.

Eine ODM-Datei, die eine Menge binärer Daten (als base64 codiert) enthält, kann sehr groß werden, was später bei der Darstellung und beim Signieren mit einer XML-Signatur zu Problemen führen kann (siehe weiter unten).

Für die Gesamtgröße des Archivs besteht kein wirkliches Problem: während das CDISC-Logo als gif-Datei eine Dateigröße von 6 kB (165x57 Pixel) hat, wächst die Größe der entsprechenden Textdatei (base64 codierte Zeichen) auf 8 kB. Ein anderer Nachteil besteht bei der Darstellung; die base64-Datei muss zuerst wieder in das ursprüngliche Binärformat transformiert werden.

Es existiert eine zweite Methode, die das neue „ItemDataURI“-Element einsetzt. Es ermöglicht einen Datenpunkt, der ein „unique resource identifier“, z.B. ein URL („unique resource locator“) ist, hinzuzufügen. Beispiel:

```
<ItemDataURI ItemOID="IT.CDISCLOGO">../images/newlogo.gif</ItemDataURI>
```

Hier wird gezeigt, dass sich die Logodatei im Bilderverzeichnis befindet, das ein Nebenverzeichnis (Sibling) des aktuellen Verzeichnisses ist.

Für die Inhalte des URI Wertes soll man sich an die W3C Notation halten, z.B. soll Datei „file://“ für Dateien, „ftp://“ für die FTP-Orte und „http://“ für die Web-Orte benutzt werden.

Für den Inhalt des „xlink:href“ Attributs wird empfohlen, die W3C Notation zu benutzen, d. h. „file://“, für Dateien „ftp://“ für ftp-Adressen und „http://“ für Web-Adressen zu verwenden<sup>16</sup>.

<sup>16</sup> Der erste Teil „http“, „ftp“, „file“ wird auch als „Protokoll“ bezeichnet.

Wenn die externen Daten auf dem gleichen Dateisystem wie das Archiv gespeichert sind, kann man selbstverständlich auch relative Dateipfade benutzen, z. B.:

```
<ItemDataExternal ItemOID="IT.CDISCLOGO" xlink:href="../../images/newlogo.gif" />
```

#### 4.7. XML-Signatur

Ab CDISC ODM v.1.2 ist die (optionale) Hinzufügung einer XML-Signatur [<http://www.w3.org/TR/xmldsig-core/>] am Ende des ODM-Dokuments (als letztes Element) Teil des CDISC-ODM-Standards.

XML-Signatur ist ein offener W3C XML-Standard. Anstatt zu versuchen, das Rad neu zu erfinden, hat sich das ODM Team entschieden, einen existierenden W3C-Standard in ODM zu implementieren.

Die wichtigste Anwendung der XML-Signatur ist sicherlich seine Verwendung in Web-Diensten, die zum Beispiel auf SOAP basieren, wo sie für die Sicherung der Datenintegrität und zur Authentifizierung benutzt wird.

XML-Signaturen dienen zwei Zwecken:

- die Integrität des Inhalts der XML-Datei (oder eines Teils davon) durch eine Übersicht sicherzustellen
- die Inhalte mit einer elektronischen Unterschrift „signieren“ zu können, um die Authentifizierung der unterzeichnenden Person sicherzustellen

XML-Signaturen können nicht nur auf XML-Dokumente (oder Teile von ihnen) angewendet werden, sondern auf beliebige elektronische Inhalte, wie zum Beispiel Bilder, Audio-Dateien usw. In diesen Fällen wird die XML-Signatur in einer separaten Datei gespeichert (detached XML-Signatur).

Grundsätzlich kann man, wenn man vom Ort, an dem die XML-Signatur gespeichert wird, ausgeht, drei Arten von XML-Signaturen unterscheiden:

- eingeschlossene (enveloped) XML-Signatur: die Signatur wird in dem XML-Element platziert, das signiert wird
- einschließende (enveloping) XML-Signatur: der signierte Inhalt wird in der XML-Signatur platziert
- anhängende (detached): Die XML-Signatur wird separat vom unterzeichneten Inhalt gespeichert.

Der CDISC-ODM-Standard unterstützt nur die „enveloped“ XML-Signatur und diese nur auf der Ebene des obersten ODM-Elements. Dies bedeutet, dass nur der komplette Inhalt der Datei signiert<sup>17</sup> werden kann. Dies ist ideal für den Zweck der Archivierung als langfristige Aufbewahrung der Dokumente.

---

<sup>17</sup> Bei der Entwicklung von ODM 1.2 schlug der Autor vor, es zu ermöglichen, einzelne Elemente der ODM-Datei zu signieren, z. B. Formulare (FormData). Für diesen Vorschlag konnte allerdings im ODM-Team keine Mehrheit gefunden werden. Es ist jedoch sehr leicht, eine ODM-Erweiterung zu entwickeln, die XML-Signaturen auf jedem gewünschten Niveau erlaubt.

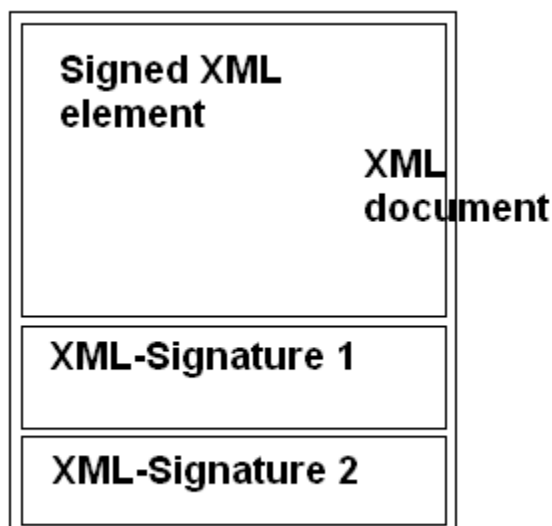


Abb. 3: Prinzip der „eingeschlossenen“ (enveloped) Signatur

Einige Beispiele von signierten ODM-Dateien werden im Anhang (Abschnitt „XML-Signature“) mitgeliefert.

Wenn eine XML-Signatur verwendet wird, stehen zwei Signatur-Ebenen zur Verfügung:

1. Mit dem Ziel, die Integrität der signierten Inhalte zu sichern, was bedeutet, dass die XML-Signatur ungültig wird, wenn sich ein einzelnes Zeichen in dem Inhalt ändert, für den die Signatur erstellt wurde.
2. Mit dem Ziel der Authentifizierung, um sicher zu stellen, dass es möglich ist, die Identität der Person zu verifizieren, die den Inhalt signiert hat.

Die erste Ebene ist einfach zu implementieren, da es zahlreiche Softwarebibliotheken gibt, die benutzt werden können, um XML-Signaturen zu erzeugen. Einige XML-Editoren verfügen über eine Funktion, die es erlaubt, auch Teile des XML-Dokuments zu signieren<sup>18</sup>.

Die zweite Ebene erfordert, dass der Signierer ein gültiges elektronisches Zertifikat besitzt, das momentan aber noch nicht einfach zu erwerben ist. Es gibt jedoch einige aussichtsreiche Entwicklungen auf diesem Gebiet (siehe weiter unten).

Es muss hier angemerkt werden, dass beim Signieren eines XML-Dokuments, Bilder, auf die in der XML-Signatur verwiesen wird, nicht selbst signiert werden, sondern nur ihr Verweis (Link). Die Einbindung der Bilder als base64-kodierte Daten ermöglicht nur, dass das Bild im Prinzip in der XML-Datei co-signiert wird.

#### 4.8. Elektronische Zertifikate mit XML-Signatur

In Deutschland ist die Verwendung elektronischer Zertifikate für die Signierung elektronischer Dokumente (wie zum Beispiel E-Mails oder anderer Dokumente) noch nicht weit verbreitet. Ein Grund dafür liegt möglicherweise in der Dezentralisierung der

---

<sup>18</sup> Beispiele sind oXygen und der Cladonia Exchanger XML Editor.

Verantwortlichkeiten für den Datenschutz über viele regionale Verwaltungen (Bundesländer), Gemeindeverwaltungen, Finanzämter usw., was zu unkoordinierten Maßnahmen und lang andauernden Verhandlungen<sup>19</sup> führt.

Die (kommunalen) Behörden sind technisch überhaupt noch nicht bereit, elektronisch signierte Dokumente anzunehmen.

In Deutschland ist das Verfahren zum Erlangen eines elektronischen Zertifikats auch noch zu kompliziert und beinhaltet hohe Kosten.

Sobald ein Anwender/eine Organisation ein elektronisches Zertifikat von einer Zertifizierungsstelle hat, kann diese benutzt werden, um XML-Signaturen mit dem elektronischen Zertifikat zu erzeugen. Ein Beispiel ist unten abgebildet.

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
</ds:SignedInfo>
<ds:SignatureValue>
M6WBQTCvTgMU063NQC48sUDxcjn5K0m0JF8zRx0Uau/c+2Wsfdr6CKCEOK12PiW3ig/x1DkIcMyf
c5yIhYK1Dtot/hGdNydp+ZpkQa4eu/7rN20psKRhy9e00VHOieZC2D85SX/qZ79f1ZTxQwynwQho
9vnfEFKcl34g5kcaUxw=
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509SubjectName>CN=Exchanger,OU=Development,O=Cladonia,C=IE</ds:X509SubjectName>
<ds:X509Certificate>
MIICBjCCAW8CBEBM1o8wDQYJKoZIhvcNAQEEBQAwwSjELMAkGA1UEBhMCUUXETAPBgNVBAoTCENs
YWRvbmlhMRQwEgyDVQQLewtEZlZlbG9wbWVudDESMBAGA1UEAxMJRXhjaGFuZ2VvYmB4XDTA0MDMw
ODElNTEOM1oXDTEyMDUyNTElNTEOM1owSjELMAkGA1UEBhMCUUXETAPBgNVBAoTCENsYWRvbmlh
MRQwEgyDVQQLewtEZlZlbG9wbWVudDESMBAGA1UEAxMJRXhjaGFuZ2VvYmB4XDTA0MDMw
AQUAA4GNADCBiQKBgQC/363xtXik6JrwR4vNx6xrp+00c8fP02jQif3yech80vEXnGxLyoyK+xEA
2hlJAzkzoyzSulursHYbnFwWFF/P65oYzv6BdI2aWY7g7HPLpZd4sawVPg9BbqaF4wBtTMAxRAi3
Eq16fQXG4gRpp3yaTuWgQ+ydHy2TbaE6+z/dAQIDAQAEMAOGCSqGSIb3DQEBAUAA4GBAfr9UXaK
gKv2S21T4PIZMjQePHzybpoMhq8JQQ2292foweThz2wfKTW36o+bYIdcNoIldV9zQ03sCAGxKXTh
9pZQLNjnH9jAGb70Q+ogmxjwIiA9KXD3U7eZ4kzCBEvf28Tp6/HsEs01WGBrbCl/hnvHVgZQD+En
xogOMwTfTLms
</ds:X509Certificate>
</ds:X509Data>
```

Das Zertifikat wird in das Element „ds:X509Data“ als Element „ds:X509Certificate“ eingefügt.

Ohne näher auf die technischen Details einzugehen, ein elektronisches Zertifikat ermöglicht es, die Identität des Signierers eines Dokuments festzustellen. Zur Validierung benötigt man

<sup>19</sup> In kleineren Ländern mit einer starken Zentralregierung ist dies viel leichter. Zum Beispiel in den Niederlanden erhält jede Person bei Geburt (oder Einwanderung) eine „sozialfiskalische Nummer“, die für alles benutzt wird. In Deutschland erhält ein Einwohner von jeder offiziellen Stelle, mit der er/sie zu tun hat, eine andere Kennnummer.

das elektronische Zertifikat der Zertifizierungsstelle (CA)<sup>20</sup>, die das elektronische Zertifikat ausgestellt hat.

Wenn man eine XML-Signatur einschließlich Zertifikat validiert, erhält man folgendes Ergebnis:

```
The XML-Signature is: valid

X509 Certificate:
Issuer: CN=Exchanger, OU=Development, O=Cladonia, C=IE
Certificate valid from 8 Mar 2004 15:51:43 GMT to 25 May 2012 15:51:43 GMT
Serial number: 1078761103
```

Im Bereich klinischer Forschung haben einige EDC-Systeme bereits XML-Signaturen mit elektronischen Zertifikaten für die Datenübermittlung implementiert. Aufgrund der Arbeitsweise von CAs sind die Kosten noch zu hoch, um die elektronischen Zertifikate auch bei der Datenerfassung an Studienzentren einzusetzen. Es gibt auch einige Anbieter, die XML-Verschlüsselung für die Datenübermittlung über das öffentliche Internet einsetzen. XML-Verschlüsselung basiert auf den gleichen Prinzipien wie die XML-Signatur. Die meisten Softwarebibliotheken, die XML-Signaturen implementieren, können daher auch eine XML-Verschlüsselung implementieren.

Um die Verwendung elektronischer Signaturen und Zertifikate in der Pharmabranche zu erleichtern und zu fördern, wurde durch die Pharmabranche 2004/2005 eine Non-Profit-Organisation gegründet: SAFE-BioPharma [[www.safe-biopharma.org](http://www.safe-biopharma.org)]. SAFE steht für „Signatures And Authentication for Everyone“ (Signaturen und Authentifizierung für alle). Sie entwickelt mehrere Standards<sup>21</sup> und ermöglicht es Organisationen und Firmen im Pharma-Umfeld, elektronische Zertifikate für Personen und Systeme zu erhalten. Das elektronische Zertifikat wird üblicherweise auf einem USB-Stick oder auf Smartcards (also auf Hardware) gespeichert und wird durch Benutzernamen und Kennwort geschützt.

SAFE könnte tatsächlich die Verwendung elektronischer Signaturen und Zertifikate in der Pharmabranche fördern. Man könnte sich vorstellen, dass eCRFs durch Prüfarzte mit ihrem SAFE-Zertifikat unterzeichnet werden und diese als signiertes XML (nicht unbedingt ODM) an den EDC-Server geschickt werden.

CDISC und ODM arbeiten momentan mit SAFE zusammen, um die Verwendung von elektronischen Signaturen und Zertifikaten bei EDC und der Archivierung von ODM (und anderen XML-Dokumenten) zu einer verbreiteten Vorgehensweise zu machen.

#### 4.9. Verwendung der XML-Signatur bei Archivierungsprojekten mit ODM

Bei der Archivierung von klinischen Studien im ODM-Format ist die XML-Signatur eine hervorragende Maßnahme, um die Integrität der archivierten Informationen sicherzustellen.

---

<sup>20</sup> Es ist auch möglich, seine eigenen Zertifikate (selbst-signierte Zertifikate) zu erzeugen. In diesem Fall kann das Zertifikat nur gegen sich selbst validiert werden, was selbstverständlich kein Beweis der wirklichen Identität des Unterzeichners ist.

<sup>21</sup> Um die europäischen Rechtsvorschriften zu erfüllen, wird die elektronische Unterschrift wahrscheinlich auf dem XAdES-Standard aufbauen, der eine Erweiterung von XML-Signature ist. Dies könnte bedeuten, dass in Zukunft der ODM-Standard XAdES statt XML-Signature implementieren muss.

Softwarebibliotheken für das Signieren von XML-Dokumenten sind verfügbar (freie, Open Source oder kommerziell), zum Beispiel:

Java Standard Edition 6 verwendet XML-Signature

[\[http://java.sun.com/developer/technicalArticles/xml/dig\\_signature\\_api/\]](http://java.sun.com/developer/technicalArticles/xml/dig_signature_api/), die Apache Foundation hat ihr Santuario Projekt (für Java und C++), und C# und VB.Net Entwickler können die Klassen aus dem Namensraum System.Security.Cryptography.Xml verwenden.

Es müssen jedoch einige Punkte beachtet werden:

- Beim Signieren von XML-Dokumenten mit der XML-Signatur (aber auch während der Validierung) wird der komplette XML-Baum in den Arbeitsspeicher geladen. Dies bedeutet, dass ein Arbeitsspeicher benötigt wird, der mindestens 10-20-mal größer ist als die Dateigröße des ODM-Dokuments. Wenn man also eine XML-Signatur verwenden möchte, sollte man darauf achten, dass die erzeugten ODM-Dateien nicht zu groß sind.
- Nach dem Signieren muss man darauf achten, dass sich die Dateiinhalte nicht ändern, da sonst die Signatur ungültig wird. Denken Sie daran, dass sogar durch kleine Änderungen die Signatur ungültig werden kann und eine Zeilenschaltung oder ein Tabulatorvorschub dies sicherlich zur Folge hat. Deshalb führt zum Beispiel eine Formatierung der XML-Inhalte, um die Elemente entsprechend der Baumstruktur eingerückt darzustellen, nach dem Signieren dazu, dass die XML-Signatur ungültig wird.
- Dies kann ein Problem darstellen, wenn aus archivierten XML-Dokumenten ein Data Warehouse erstellt wird. Falls diese, zum Speichern der Dokumente im Data Warehouse, zerlegt werden müssen, muss man darauf achten, dass sie beim Exportieren aus dem Data Warehouse auf genau die selbe Art und Weise wieder aufgebaut werden, wie sie ursprünglich aufgebaut gewesen sind – sicherlich keine leichte Aufgabe.  
Daher bin ich persönlich für die Verwendung von nativen XML-Datenbanken, da sie es erlauben, XML-Dokumente im genau gleichen Format abzurufen, wie sie gespeichert wurden.

Einige Personen haben sich Gedanken über die Verwendung eines Web-Dienstes (z. B. basierend auf SOAP) für das Signieren von ODM Archivdateien gemacht. Das ist tatsächlich möglich, weil ein solches System auch über ein elektronisches Zertifikat (d. h. das seiner Organisation) verfügen kann. Hier könnte es mit der Menge an Daten, die über das Internet (Traffic) gesendet werden muss, ein Problem geben: Archivdateien können schon ziemlich groß werden, weshalb lange Antwortzeiten zu erwarten sind. Dies kann selbstverständlich durch Verwenden eines Batch-basierten Web-Dienstes gelöst werden, wodurch allerdings neue Probleme entstehen.

## 5. Nach dem Archivieren

### 5.1. Tools für das Arbeiten mit ODM-Archiven

Bis heute ist die Zahl von öffentlich verfügbaren Software Tools für die Bearbeitung von ODM-basierten Archiven mit klinischen Daten noch sehr begrenzt. Für die meisten Archivierungsprojekte wurden proprietäre Tools entwickelt, um die erstellten Archive zu verwalten und durchzusehen. Selbstverständlich besteht eine einfache Lösung zum Durchsehen von ODM-Dateien in der Verwendung eines XSLT-Stylesheets, mit dem sich die



ODM-Daten in HTML-Tabellen transformieren lassen. Alternativ kann ein paralleles Archiv von PDFs erstellt werden, entweder direkt aus den klinischen Daten oder aus den erzeugten ODM-Daten.

In der Vergangenheit hatte die PHT Corporation einen ausgezeichneten ODM-Viewer [<http://www.cdisc.org/pdf/CDISCReadWriteODM27.pdf>] für die erste Version von ODM, aber leider wurde dieser nie an Version 1.1 oder höher des Standards angepasst. Daher hat der Autor seinen eigenen ODM-Viewer entwickelt, der sowohl Version 1.2 als auch 1.3 des Standards abdeckt und käuflich zu erwerben ist.

Auch steht ein CDISC ODM-Checker zur Verfügung. Momentan funktioniert er nur mit Version 1.2 des Standards. Er ist für CDISC-Mitglieder frei verfügbar<sup>22</sup>.

Es stellt sich zudem die Frage, wie Queries für ein ODM-basiertes Archiv erstellt werden können. Obwohl es der üblichen Praxis entspricht, pro Studie und Probanden zu archivieren (z. B. mit Studien-ID und Probanden-ID als Teil des Dateinamens), ist dies unpraktisch, wenn man Daten von Probanden oder sogar Studien vergleichen will.

Obwohl es noch nicht der üblichen Praxis entspricht, besteht eine sehr gute Lösung dafür darin, die XML-Dokumente<sup>23</sup> in einer nativen XML-Datenbank zu speichern. Der Vorteil dieses Vorgehens besteht darin, dass die XML-Signaturen innerhalb der XML-Dokumente ihre Gültigkeit behalten (da XML nicht zerlegt wird). Der Nachteil besteht darin, dass die Kenntnisse über die Query-Sprache für Abfragen der XML-Datenbanken (XQuery) bei Programmierern noch nicht weit verbreitet sind. Ich würde sagen, dass sich unter 100 SQL-Spezialisten nur 1 XQuery-Spezialist findet.

Native XML-Datenbanken sind in unterschiedlichen Konfigurationen und Preisklassen erhältlich. Die bekannteste ist eXist von der Apache Foundation, ist Open Source und eine frei verfügbare native XML-Datenbank [<http://exist.sourceforge.net/>]. Hier ist ein Überblick über native XML-Datenbanken, mit Verweisen auf andere Übersichten: [[http://en.wikipedia.org/wiki/XML\\_database](http://en.wikipedia.org/wiki/XML_database)].

Eine andere Lösung besteht selbstverständlich darin, die XML-Funktionen der klassischen (relationalen) Datenbanken zu verwenden. Die meisten Anbieter von relationalen Datenbanken bieten inzwischen auch Funktionen an, die es erlauben, XML-Dokumente effektiv (also nicht als BLOB) zu speichern. Oracle-Anwender (aber auch andere) können einen Blick auf Berkeley DB, eine Open Source native XML-Datenbank werfen (DB XML ist eine Ebene aufgesetzt auf die DB, die eine komplette XML-native Datenbank erzeugt. Berkeley wurde kürzlich von Oracle erworben).

## 5.2. Validierung von signierten ODM-Dateien

Falls die XML-Signaturen von ODM-Dateien mehrere Jahre nach ihrer Erstellung validiert werden sollen, muss man beachten, dass elektronische Zertifikate eine begrenzte Gültigkeit

---

<sup>22</sup> In Zusammenarbeit mit dem Autor und dem KKS-Heidelberg wurde ein völlig neuer ODM-Checker für ODM v.1.3-Dateien entwickelt, der der CDISC Community frei zur Verfügung gestellt werden soll. Die Entwicklung ist jedoch noch nicht abgeschlossen.

<sup>23</sup> Beachten Sie den Gebrauch des Wortes „Dokumente“ anstatt „Dateien“. Eine native XML-Datenbank verwendet keine Datei zum Speichern eines XML-Dokuments.



haben, das heißt, ihre Gültigkeit läuft ab<sup>24</sup>. Dies bedeutet, dass bei der Überprüfung der Authentifizierung des Signierers möglicherweise das Ergebnis besagt, dass das elektronische Zertifikat ungültig geworden ist. Dies ist kein größeres Problem, wenn es einem bekannt ist.

Außerdem gibt es bei der Validierung der XML-Signatur einer ODM-Datei, wenn die Signatur als ungültig angezeigt wird (auf Grund eines falschen Digests), kein Verfahren um herauszufinden, warum die Unterschrift ungültig ist. Es kann schon daran liegen, dass jemand die Datei bearbeitet hat, irgendwo nur einen Leerschritt oder eine Zeilenschaltung hinzugefügt hat und die klinischen Daten überhaupt nicht geändert hat. Dennoch ist dadurch die Signatur ungültig geworden und den klinischen Daten kann im Prinzip nicht mehr vertraut werden.

### 5.3. Archivierung und Gute Klinische Praxis (GCP)

#### 5.3.1. Beziehung zum Trial Master File (TMF)

Die Pflege eines Trial Master Files (TMF), der alle essentiellen Dokumente einer Studie enthält, ist eine dokumentenbasierte Bedingung Guter Klinischer Praxis (GCP). In Beziehung zum TMF kann man die klinische Datenbank nur als ein Element des vollständigen TMFs ansehen. Mit der Einführung eines Dokumentenmanagementsystems durch die Mitglieder der Telematik-Plattform sollte man für die Zukunft die vermehrte Erstellung von elektronischen Studiendokumenten und folglich von elektronischen TMFs in Kombination mit elektronischen Signaturen erwarten.

Das CDISC ODM-Archiv ist nur ein Teil des TMFs. Dieser wird noch viel mehr (elektronische) Dokumente enthalten. Auch diese können mit einer elektronischen Signatur (und Zertifikat) signiert werden, auch wenn dies keine XML-Signatur<sup>25</sup> zu sein braucht.

Andere elektronische Dokumente wie PDFs können mit anderen Technologien (einige proprietär, einige offen) elektronisch unterzeichnet werden. Im Prinzip können die gleichen elektronischen Zertifikate verwendet werden, um sowohl PDFs als auch XML-Dokumente zu signieren. So können zum Beispiel die elektronischen Zertifikate von SAFE-BioPharma in Verbindung mit Adobe Software verwendet werden, um PDF-Dateien elektronisch zu signieren.

### 5.4. Fazit: Die langfristige Stabilität von CDISC-ODM für die Archivierung

Da ODM auf XML basiert und XML momentan das internationale Format der Wahl für den Datentransport und -austausch ist, bin ich überzeugt, dass XML auch die nächsten 20-30 Jahre zur Verfügung stehen wird. Dies bedeutet, dass XML-Archive deshalb im Prinzip auch in 20-30 Jahren noch lesbar sein werden.

Man kann darüber diskutieren, ob CDISC und ODM in 20-30 Jahren noch existieren werden. Dies ist natürlich nicht so leicht vorherzusagen.

Es gibt einige Tendenzen innerhalb CDISC, sich in Richtung HL7-XML Messages zu bewegen, aber dies ist sehr strittig. Meine persönliche Meinung ist, dass dies eine sehr schlechte Entscheidung wäre<sup>26</sup>.

---

<sup>24</sup> Dies ist einer der Gründe, warum SAFE sich für XAdES entschieden hat, da es ermöglicht, die Signatur zu validieren, wenn das Zertifikat abgelaufen ist, d. h. man kann validieren, ob die Signatur zu der Zeit gültig war, als das Dokument signiert wurde.

<sup>25</sup> SAFE Signaturen können momentan zu PDF- und Word-Dokumenten hinzugefügt werden

Selbst wenn eines Tages für die Archivierung klinischer Daten von ODM-XML auf HL7-XML umgestiegen werden sollte, können ODM-Archive leicht in HL7-XML-Archive umgewandelt werden. Es könnte dann natürlich ein Problem für ODM-Erweiterungen entstehen.

Ein Problem wird selbstverständlich immer die Stabilität der Speichermedien sein. Die Lebensdauer einer CD oder DVD ist sehr abhängig vom Hersteller, der Art der Aufbewahrung (Licht, Temperatur, Luftfeuchtigkeit) und sehr schwierig vorherzusagen.

Wenn man also auf CD/DVD archiviert, sollte man einen Prozess einrichten, um die Inhalte und Integrität der Dateien auf den Medien regelmäßig (vielleicht jährlich) zu prüfen und um Backups zu erstellen. Zum Beispiel ist es keine schlechte Idee, Medien von verschiedenen Herstellern für verschiedene Backup-Serien zu verwenden.

Um schnell den Inhalt einer Studie „wiedergeben“ zu können, ohne das ganze Archiv installieren zu müssen, würde ich vorschlagen, eine native XML-Datenbank zu verwenden. Obwohl die Erstellung von Queries für native XML-Datenbanken noch keine Fähigkeit ist, die von vielen Datenbankmanagern beherrscht wird, wird XQuery an Fachhochschulen und Universitäten gelehrt, und man kann erwarten, dass die nächste Generation der Datenbankmanager auch über die erforderlichen Fähigkeiten verfügt. Der Vorteil besteht darin, dass man bei Verwendung einer nativen XML-Datenbank sicher sein kann, dass der Inhalt 100 % identisch mit den Inhalten des Archivs ist. Eine ordnungsgemäß eingerichtete native XML-Datenbank erleichtert es, bei Durchsichten Fragen zu speziellen Probanden von Studien (oder Vergleiche zwischen Probanden) zu beantworten, da keine Kenntnisse der Datenbankstruktur, sondern nur von ODM und von XQuery erforderlich sind (da sie nur als „Behälter“ für ODM-Dokumenten dient).

## 6. Anhang

### 6.1. Literatur

Wheeldon, 2008: Using the ODM to optimise Study Set-up and EDC System Build within a Multi-vendor Environment:

[http://www.cdisc.org/publications/euinterchange2008/day1/session6\\_2/MarkWheeldonS6Tk2.pdf](http://www.cdisc.org/publications/euinterchange2008/day1/session6_2/MarkWheeldonS6Tk2.pdf)

Aerts, 2006: CDISC Standards energize EDC, Applied Clinical Trials, October 2006:

<http://appliedclinicaltrialsonline.findpharma.com/appliedclinicaltrials/article/articleDetail.jsp?id=377747>

Cassels, 2001: The Data Archiving Challenge: <http://www.cdisc.org/pdf/Archive0522ppt.pdf>

Kubick, 2005: Case Study: Applying CDISC Standards to Replace a Legacy Clinical Data Management System: [http://www.cdisc.org/pdf/Interchange2005/18-KubickCDISC\\_Standards\\_Case\\_Study\\_Euro.pdf](http://www.cdisc.org/pdf/Interchange2005/18-KubickCDISC_Standards_Case_Study_Euro.pdf)

Forrester, 2005: Robert Markham: XML and PDF-A will Reign as Future Document Archive Formats: <http://www.forrester.com/Research/Document/Excerpt/0,7211,36355,00.html>

Häber, 2008: XML als Format zur elektronischen Archivierung von Studiendaten. Draft version of expert report V042-01




---

<sup>26</sup> Ich habe kürzlich eine Publikation geschrieben, die demnächst veröffentlicht wird: „Zehn gute Gründe, warum HL7-XML nicht immer eine gute Lösung als ein Format für CDISC-Standards ist“.












SAS, TS140: The Record Layout of Data Sets in SAS Transport (XPORT) Format:  
<http://support.sas.com/techsup/technote/ts140.html>

## 6.2. Dateien der Begleit-CD zum Gutachten


### 6.2.1. XML-Signatur

- CTChicago\_XSchema\_metadataonly\_signed.xml 
- CTChicago\_XSchema\_small\_signed.xml 
- MetaData\_Dave\_1\_3\_new\_2006\_01\_26\_signed.xml 



### 6.2.2. CDISC ODM Vendor Extensions

- ODM\_1-2-1\_vendor\_extensions.pdf 
- my\_simple.xml 
- StudyDesign\_with\_extensions.xml 
- my\_simple.xsd 
- my\_simple-extension.xsd 
- my\_simple-ns.xsd 
- ODM1-3-0.xsd 
- ODM1-3-0-foundation-draft.xsd 
- xlink.xsd 
- xml.xsd 
- xmldsig-core-schema.xsd 

### 6.2.3. Sample

- zbi.net-2004-02-27-oncology002.xml 

### 6.2.4. Literature

- 18-KubickCDISC\_Standards\_Case\_Study\_Euro.pdf 
- Cassels\_Archive\_2001.pdf 
- MarkWheeldonS6Tk2.pdf 