

NINA HOCHEDLINGER, BSc

**Standardisierter Datenaustausch in medizinischen
Forschungsnetzwerken mit Anwendung in der
pädiatrischen Onkologie**

MASTERARBEIT
zur Erlangung des akademischen Grades
Diplom-Ingenieurin
Masterstudium Biomedical Engineering

eingereicht an der
TECHNISCHEN UNIVERSITÄT GRAZ

BETREUER:
Univ.-Doz. DI Dr. techn. Günter Schreier, MSc
Institut für Semantische Datenanalyse/Knowledge Discovery

27. April 2015

IN ZUSAMMENARBEIT MIT



AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH

Digital Safety and Security Department

Reininghausstraße 13/1

8020 Graz

Österreich

BETREUER:

DI Dr. Dieter Hayn

This work was partly funded by the European Commission under FP7-HEALTH-F2-2011 Contract no. 261474 (ENCCA - European Network for Cancer Research in Children and Adolescents).



Zusammenfassung

Klinische Studien spielen auf dem Gebiet der pädiatrischen Onkologie eine große Rolle. Daten, die im Rahmen von klinischen Studien erhoben wurden, sind oft auch wichtig für weitere wissenschaftliche Fragestellungen außerhalb der Studie. Insbesondere bei Patienten, die in mehr als einer klinischen Studie behandelt wurden, oder für die Daten aus Biopsien in Biobanken abgelegt wurden. Um diese Szenarien abbilden zu können, müssen Daten aus unterschiedlichen Quellen miteinander verknüpft werden. Voraussetzung dafür ist ein einheitlicher Datenstandard. In dieser Arbeit wurde ein Tool entwickelt, mit dem Daten aus einem klinischen Studiensystem in das standardisierte Operational Data Model (ODM) exportiert werden können. Um semantische Interoperabilität zu erreichen, wurde ein weiteres Tool implementiert, das durch Einsatz von Extensible Stylesheet Language Transformations (XSLT) vordefinierte Kerndatensätze aus den Studien-Export-Dokumenten extrahiert. Beide Tools wurden erfolgreich im Rahmen eines Showcases demonstriert. Durch diese beiden Tools können bestehende Datenbestände für die Zweitverwendung (Secondary Use) zielgerichtet aufbereitet werden. Sie tragen damit dazu bei, wichtige und bisher schwer bearbeitbare Fragestellungen aus der pädiatrischen Onkologie und darüber hinaus zu lösen.

Abstract

In paediatric oncology clinical trials play a particularly important role to better understand the diseases and to find more efficient treatment strategies. Data that were collected in the course of clinical trials are potentially valuable for additional scientific research questions in so called secondary use scenarios. Examples are questions about patients who participated in more than one clinical trial, or questions that are related to a patient's biosamples, that are usually stored in different databases. In all those cases, the collected data need to be connected. For the data analysis from different sources, the use of standards is necessary. In this work a tool was developed that allows the export of data from the relational database of a clinical trial into the standardized Operational Data Model (ODM). For semantic interoperability, a tool was developed which extracts predefined core datasets from the study export files based on Extensible Stylesheet Language Transformations (XSLT). Both tools were successfully demonstrated in a showcase. The documents are thus prepared for further data analysis, which could result in new findings in pediatric oncology and beyond.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Pädiatrische Onkologie	1
1.2. Medizinische Forschungsnetzwerke	2
1.2.1. Usecase 1 - Horizontale Integration	3
1.2.2. Usecase 2 - Vertikale Integration	4
1.3. ENCCA	6
1.4. Aufgabenstellung	8
2. Stand der Technik	9
2.1. Forschungsportale	9
2.2. CDA	10
2.3. ODM	12
2.3.1. ODM-Element Study	13
2.3.2. ODM-Element ClinicalData	17
2.3.3. Konformität	17
2.4. CDISC ODM Checker Tool	18
2.5. LTI-Studiensystem	19
2.6. ABCD-4-E	21
2.6.1. Patient Registration App und Pseudonymisierung	24
2.6.2. Data Management App	25
2.7. ENCCA-Kerndatensatz	27
2.8. XSLT	29
3. Methoden	32
3.1. ODM-Export	32
3.1.1. Fall 1, nicht wiederkehrende Events	32
3.1.2. Fall 2, wiederkehrende Events	34
3.1.3. Exportfunktion LTI-Studie	34

3.1.4. Ablauf	36
3.1.5. Python Skript	37
3.1.6. Validierungsfunktion	39
3.2. ODM-Transformation	40
3.2.1. Kerndatensatz	40
3.2.2. Data Management App	43
4. Ergebnisse	45
4.1. ODM-Export	45
4.2. ODM-Transformation	47
5. Diskussion	53
5.1. Outlook und Future Work	56
5.2. Schlussfolgerung	57
Abkürzungsverzeichnis	58
Literatur	60
A. XSLT Stylesheet	65

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am
.....
(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date
.....
(signature)

1. Einleitung

1.1. Pädiatrische Onkologie

Krebs zählt zu den häufigsten Todesursachen weltweit. 2011 erkrankten in Österreich 38.695 Menschen an Krebs. Im Jahr 1983 waren es im Vergleich 29.344 Neuerkrankungen. Laut Prognose soll die Inzidenz bis zum Jahr 2030 auf 43.706 ansteigen [1].

Moderne Behandlungsmethoden wie Operationen, Chemotherapien, Immuntherapien, Strahlentherapien und Kombinationen daraus werden durch die stetige Forschung kontinuierlich weiterentwickelt. Durch neue Erkenntnisse im Bereich der Tumorbilogie, durch verbesserte Diagnostik und durch neue, zielgerichtete Therapieformen konnten Fortschritte in der Früherkennung, Behandlung und Prävention in den letzten Jahren vorangetrieben werden [2]. Die 5-Jahres-Überlebensrate stieg von 67% (1987) auf 86% (2007) [3].

Etwa 1% aller Krebsdiagnosen tritt bei Kindern und Jugendlichen auf [4]. Jährlich wird in Europa bei ca. 15.000 Kindern (0-14 Jahre) und 20.000 Jugendlichen (15-24 Jahre) Krebs diagnostiziert [5]. Die 5-Jahres-Überlebensrate lag bei den zwischen 2000 und 2007 diagnostizierten Patienten bei 77,9% [6]. Trotz der Fortschritte in der Kinderkrebsforschung sterben jährlich rund 3.000 von den genannten 15.000 Kindern in Europa an den Folgen ihrer Erkrankung [7].

Neuroblastom ist mit einem Anteil von ca. 7-8% aller Neuerkrankungen im Kindesalter eine der häufigsten pädiatrischen Tumorerkrankungen [8]. Weitere häufige Krebsarten sind Leukämien, Wilms-Tumore, Hirntumore, Rhabdomyosarkome, Lymphome, Retinoblastome, Osteosarkome und Ewing-Sarkome.

Die Evidenz aus der Forschung mit erwachsenen Krebspatienten kann für die pädiatrische Onkologie nicht verallgemeinert werden. Es besteht daher ein dringender Bedarf für qualitativ hochwertige Forschung in diesem Bereich, um sicherzustellen, dass Medikamente,

die in der pädiatrischen Onkologie verwendet werden, auf die Bedürfnisse von Kindern und Jugendlichen abgestimmt sind [9]. Viele Kinder werden daher in klinischen Studien behandelt. Durch die Behandlung der Patienten in klinischen Studien wird zum einen gewährleistet, dass die Kinder eine dem Stand der Forschung entsprechende Therapie erhalten, zum anderen soll durch neue Erkenntnisse und daraus resultierende, verbesserte Therapien der Outcome verbessert werden.

Die oben gelisteten, statistischen Zahlen sind einerseits als hoch anzusehen, da sie gerade im pädiatrischen Umfeld mit viel Leid für Kinder, Eltern, Familienangehörige und Freunde verbunden sind. Andererseits ist die Inzidenz der Krankheit für viele wissenschaftliche Fragestellungen relativ niedrig, und es ist ein zeitintensiver Prozess, um aus den klinischen Studiendaten statistisch signifikante Forschungsergebnisse zu erzielen und somit neue Behandlungsmöglichkeiten zu etablieren. Um eine entsprechende Menge an Patientendaten in einer angemessenen Zeit erfassen zu können, ist daher eine internationale Kooperation zwischen den einzelnen pädiatrisch-onkologischen Einrichtungen in Europa erforderlich [10]. Zahlreiche multizentrische klinische Studien im Bereich der pädiatrischen Onkologie in Europa sind das Ergebnis dieser Kooperationen, wodurch trotz geringer Inzidenz ausreichend hoher Erkenntnisgewinn in vertretbarer Zeit ermöglicht wird. [11].

1.2. Medizinische Forschungsnetzwerke

Die International Society of Paediatric Oncology Europe Neuroblastoma (SIOPEN) ist eine internationale Expertengruppe, die auf die Forschung und die Verbesserung der Behandlung von Patienten mit Neuroblastom spezialisiert ist [12].

Das Forschungsnetzwerk International Society of Paediatric Oncology European Neuroblastoma Research Network (SIOPEN-R-NET) ist ein webbasiertes IT-Netzwerk für die interdisziplinäre biomedizinischen Forschung [13]. Insbesondere werden multizentrische klinische Studien im Bereich der Neuroblastomforschung unterstützt [10].

SIOPEN führt eine Reihe von klinischen Studien durch. Diese unterscheiden sich im Grad der Erkrankung und in der Art und der Dauer der Behandlung [14]. Aktuell laufen folgende Studien parallel ab:

- High Risk Neuroblastoma Study (HRNBL)

- Low and Intermediate Risk Neuroblastoma European Study (LINES)
- Opsoclonus Myoclonus Syndrome/Dancing Eye Syndrome in Children With and Without Neuroblastoma (OMS/DES)
- Long Term Continuous Infusion ch14.18/CHO Plus s.c. Aldesleukin (IL-2) Study (LTI)
- BEACON-Neuroblastoma Trial (BEACON) (bei refraktärem oder rezidivierendem Neuroblastom)

Für den Einschluss eines Patienten in eine klinische Studie muss dieser üblicherweise gewisse Ein- und Ausschlusskriterien erfüllen. Um abhängig vom konkreten Status eines Patienten aus einer Vielzahl parallel laufender, multizentrischer Studien die individuell am besten passende Studie zu identifizieren, wurde im SIOPEN-R-NET ein zentrales Tool zur Zuweisung von Patienten zu Studien entwickelt. Innerhalb dieses Tools wird ein Agreed Minimum Essential Data (AMED) Datensatz erfasst, der ausgewählte, häufig vorkommende Ein- und Ausschlusskriterien enthält, wie z.B. Alter oder Neuroblastom-Risikoklasse. Abhängig von den hier eingetragenen Daten wird dem Arzt eine Liste aller passenden Studien angezeigt [15]. Die Patienten werden dann vom Arzt einer bestimmten klinischen Studie zugeteilt.

1.2.1. Usecase 1 - Horizontale Integration

Nicht selten nehmen Patienten im Verlauf ihrer Krankheit an mehreren klinischen Studien teil. Ein Szenario wäre beispielsweise, dass ein Patient der Low-Risk-Gruppe initial nach dem Protokoll der LINES-Studie behandelt wird. Schlägt jedoch die Therapie nicht an, kann der Patient aus der LINES-Studie ausscheiden und er fällt damit in die High-Risk-Gruppe. In weiterer Folge erfolgt daher die Behandlung laut Protokoll der HRNBL-Studie. Auch wenn Patienten nach Abschluss einer Studie einen Rückfall erleiden, kann dies zu einer Betreuung in einer anderen Studie führen.

Des Weiteren kann es vorkommen, dass bei einem Patienten ein Wilms-Tumor diagnostiziert wird und die Behandlung daher anhand eines entsprechenden Studienprotokolls begonnen wird. Im Laufe der Behandlung könnte es sein, dass die Diagnose korrigiert werden muss, weil sich der Tumor tatsächlich als Neuroblastom herausstellt. Von da an müsste die Therapie entsprechend eines Neuroblastom-Studienprotokolls durchgeführt

werden. Beide Szenarien sind in Abbildung 1.1 dargestellt.

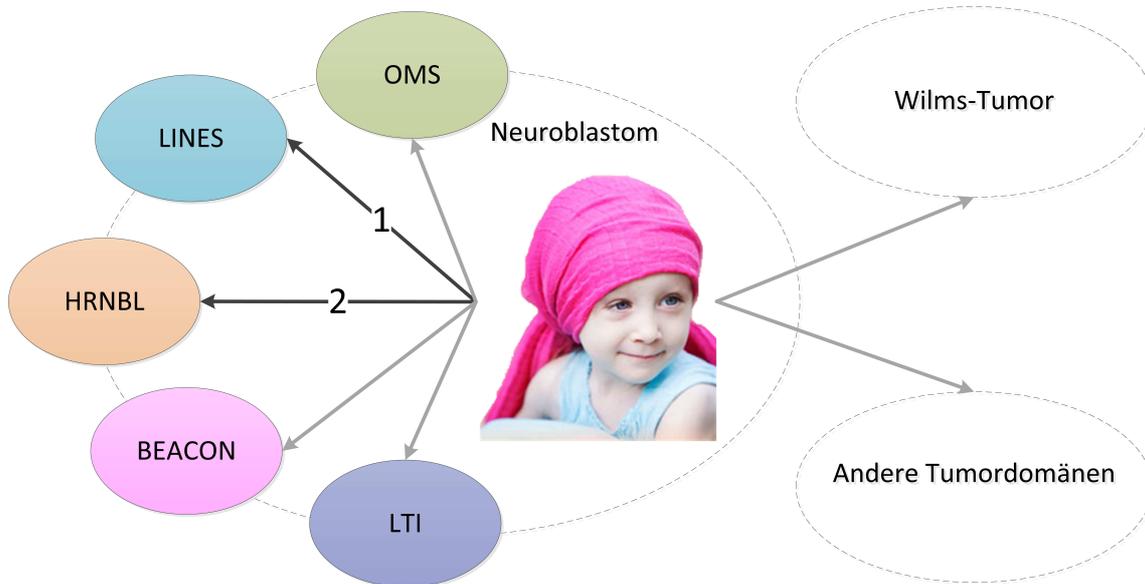


Abbildung 1.1.: Patienten werden einer bestimmten klinischen Studie zugeteilt. Ändern sich Gesundheitsparameter oder erleidet ein Patient einen Rückfall, so kann es vorkommen, dass Patienten in eine andere Studie aufgenommen werden. Des Weiteren könnten Patienten im Zuge ihrer Krankheitsgeschichte mehr als eine Krebserkrankung diagnostiziert bekommen und somit in mehr als einer klinischen Studie der jeweiligen Tumordomänen behandelt werden.

Um diese Szenarien identifizieren und eventuelle Rückschlüsse daraus ziehen zu können, müssen die klinischen Studiendaten miteinander verknüpft werden. Man spricht hier von einer horizontalen Integration von Daten (siehe Abbildung 1.2).

1.2.2. Usecase 2 - Vertikale Integration

Oft werden im Laufe der Behandlung Bioproben entnommen. Die Entnahme von Proben des Tumorgewebes kann entweder bei einer Biopsie erfolgen, oder im Zuge einer Operation - z.B. bei der chirurgischen Entfernung des Tumors. Die entnommenen Gewebeproben werden anschließend an spezialisierte Labore versandt. Beispielsweise werden Gewebeproben von Neuroblastom-Patienten in Österreich üblicherweise im St. Anna Kinderspital in Wien untersucht. Dort werden Untersuchungen zur Molekulargenetik sowie zu Mutationen (z.B. MYCN-Amplifikation) durchgeführt. Die Ergebnisse dieser

Untersuchungen sowie verbleibendes Biomaterial werden häufig in Biobanken gespeichert.

Werden klinische Studiendaten nun mit Daten aus externen Biobanken verknüpft, so handelt es sich um eine vertikale Integration.

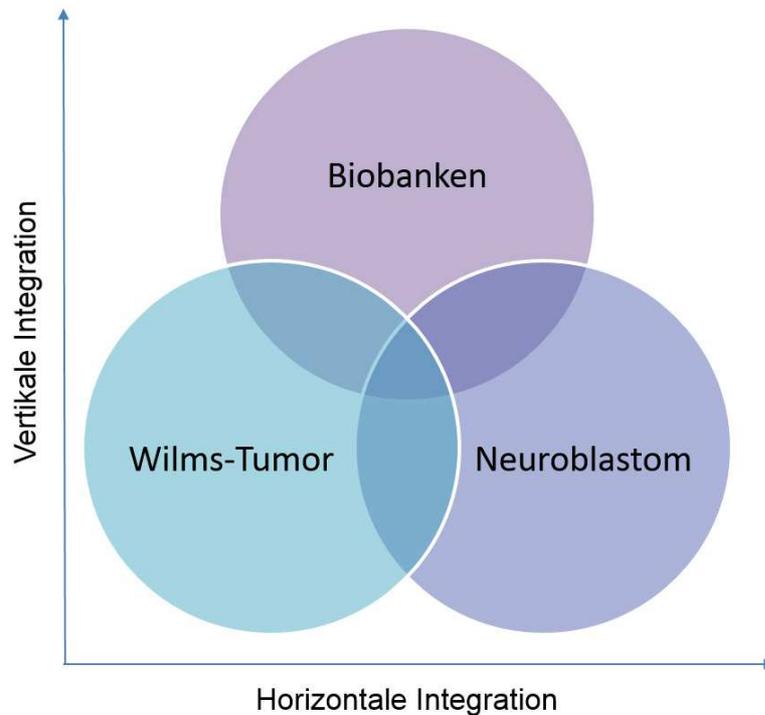


Abbildung 1.2.: Horizontale und vertikale Integration von Daten. Horizontale Integration liegt vor, wenn Daten aus unterschiedlichen klinischen Studien miteinander verknüpft werden. Werden diese Daten nun mit Daten aus externen Biobanken verknüpft, so spricht man von vertikaler Integration.

Gerade molekulargenetische Daten aus Biobanken stellen für *Personalized Medicine* eine wichtige Informationsquelle dar. Hierbei wird unter anderem auf eine Individualisierung der Art und Dosierung von Medikamenten aufgrund von verschiedenen (u.a. genetischen) Risikofaktoren gesetzt [16].

Durch eine internationale Kooperation und die Verknüpfung multizentrischer Studiendaten aus unterschiedlichen Bereichen der Forschung erhofft man sich die Beantwortung wichtiger Forschungsfragen über die bestmögliche Behandlung der Patienten und die Ursachen der Erkrankung [7].

1.3. ENCCA

Das EU-Projekt European Network for Cancer Research in Children and Adolescents (ENCCA) ist ein europäische Netzwerk der Krebsforschung für Kinder und Jugendliche und besteht aus 34 anerkannten Forschungseinrichtungen und Institutionen auf dem Gebiet der pädiatrischen Onkologie (siehe Abbildung 1.3).

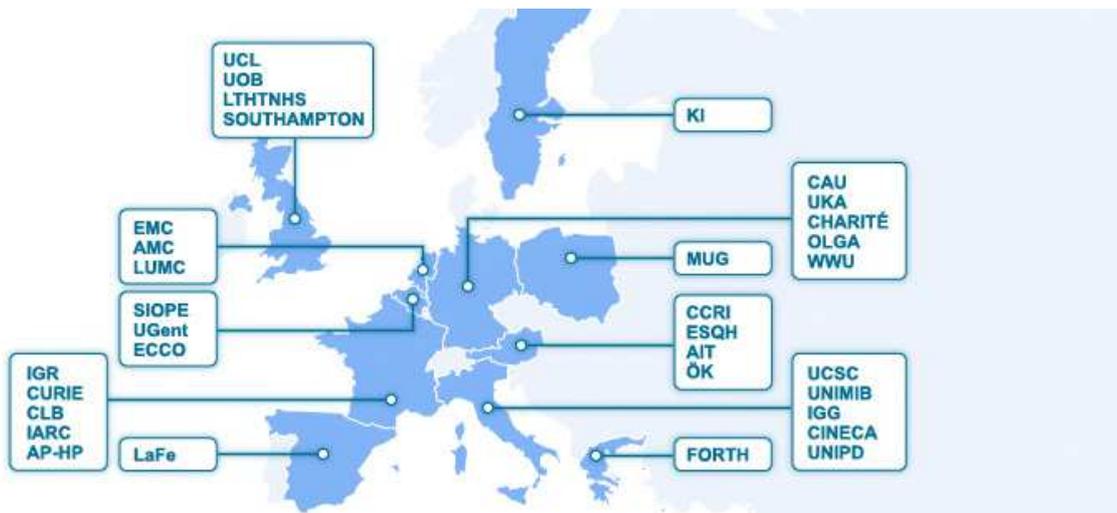


Abbildung 1.3.: Partner im EU-Projekt ENCCA [7]

Im Rahmen dieses Projekts soll durch transinstitutionale und transdisziplinäre Neuroblastomforschung der Zugang zu neuen Therapien und deren europaweite Standardisierung ermöglicht werden. Dabei soll sowohl horizontale als auch vertikale Integration, wie in Kapitel 1.2 beschrieben, realisiert werden. In Abbildung 1.4 ist die Webpräsenz als Screenshot dargestellt.

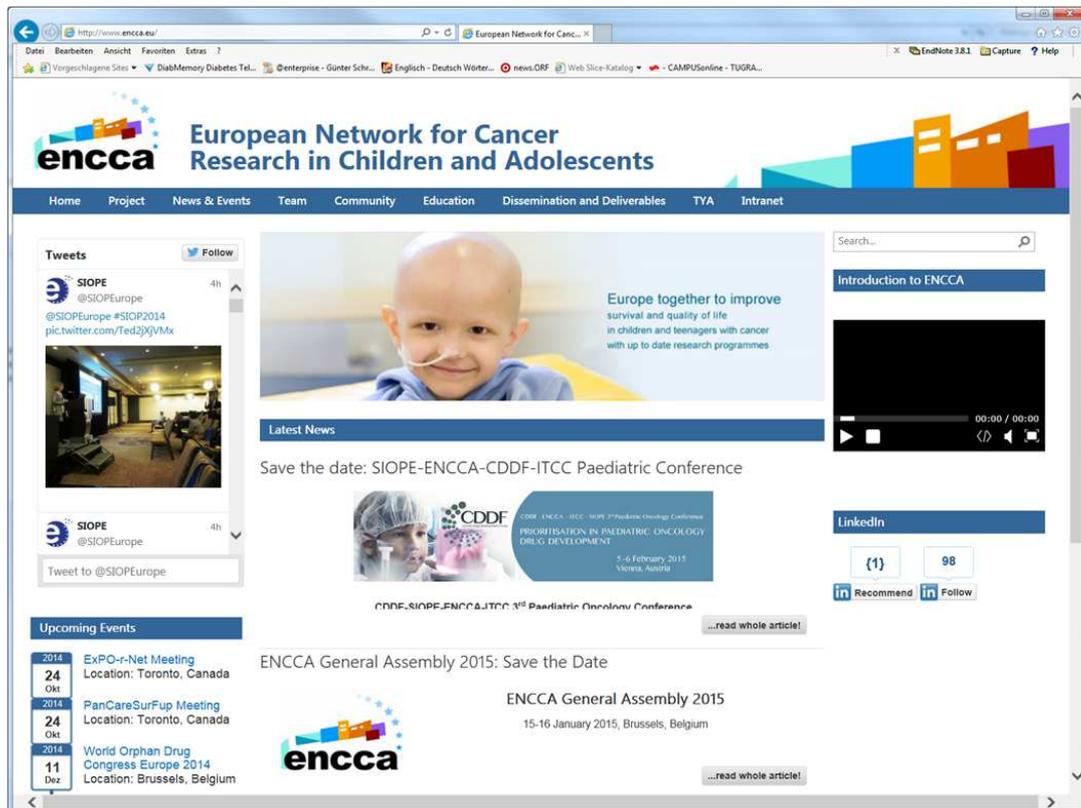


Abbildung 1.4.: Web Präsenz ENCCA [7]

Vorrangiges Ziel von ENCCA ist eine Vernetzung der Aktivitäten der europäischen Kinderkrebsforschung auf allen Ebenen. Derzeit wird eine elektronische Kommunikationsplattform für die Verknüpfung der verschiedenen Initiativen umgesetzt, genannt Advanced Biomedical Collaboration Domain for ENCCA (ABCD-4-E) [17].

Eine zentrale Komponente innerhalb der ABCD-4-E-Plattform sind kontextspezifische Pseudonyme entsprechend dem ENCCA Unified Patient Identifier (EUPID) Konzept. Durch die EUPID soll horizontale und vertikale Integration von Daten ermöglicht werden, wobei Datensicherheit und Datenschutz auch bei Studien-übergreifendem Zugriff auf pseudonymisierte Daten sichergestellt werden können [18]. Details zur Umsetzung der EUPID sind in Kapitel 2.6.1 beschrieben.

Des Weiteren wurden in ENCCA sogenannte Kerndatensätze definiert, mit deren Hilfe Daten aus unterschiedlichen Quellen miteinander vergleichbar gemacht werden sollen.

Es entsteht somit ein Pool von pseudonymisierten Daten aus unterschiedlichen Tumordomänen und Forschungsbereichen, die der sekundären Forschung verfügbar gemacht werden sollen. Von sekundärer Forschung spricht man, wenn bereits erhobene Daten zur

Analyse und Interpretation herangezogen werden.

Man verspricht sich daraus neue Erkenntnisse und ein besseres Verständnis über die Ursache und den Verlauf von Krebserkrankungen bei Kindern [7].

1.4. Aufgabenstellung

Die gegenständliche Arbeit bearbeitet eine Teilaufgabe des Gesamtprojekts ENCCA. Im Zuge der Masterarbeit soll eine Technologie entwickelt und in einer Machbarkeitsstudie getestet werden, die eine horizontale und vertikale Integration von Daten in ENCCA ermöglicht. Dazu soll sowohl syntaktische Interoperabilität (einheitliche Struktur der Daten) als auch semantische Interoperabilität (korrekte Interpretation der Daten) sichergestellt werden.

Für syntaktische Interoperabilität sollen die Daten aus der relationalen Datenbank des Neuroblastom-Studiensystems der LTI-Studie in ein standardisiertes Datenformat exportiert werden. Der Export soll aus dem bestehenden, Web-basierten, elektronischen Datenerfassungssystem der LTI-Studie getriggert werden.

Zusätzlich soll eine Web-Applikation als Teil der ABCD-4-E-Plattform implementiert werden, die aus den exportierten Daten die in ENCCA definierten Kerndatensätze extrahiert und die Daten so transformiert, dass semantische Interoperabilität sichergestellt ist.

Die transformierten Daten sollen so beschaffen sein, dass sie bereit für den Upload in ein *Repository* sind und damit der sekundären Forschung zur Verfügung gestellt werden können. Die Implementierung dieses Prozesses war jedoch nicht mehr Teil dieser Arbeit, sondern soll im nächsten Entwicklungsschritt realisiert werden.

Ein grafischer Überblick über die Aufgabenstellung ist in 1.5 dargestellt.

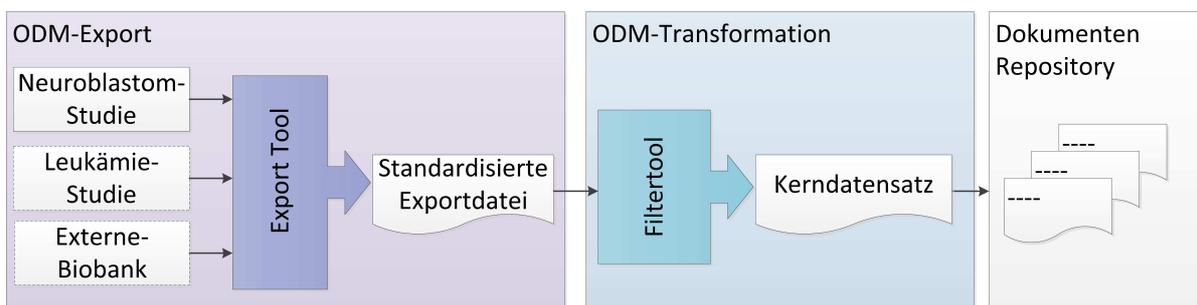


Abbildung 1.5.: Grafische Darstellung der Aufgabenstellung mit den beiden Bereichen ODM-Export und ODM-Transformation. Der Upload der Dokumente in ein *Repository* ist nicht mehr Teil der vorliegenden Arbeit.

2. Stand der Technik

2.1. Forschungsportale

Für die klinische Forschung ist eine Verknüpfung von Daten besonders interessant, da eine Analyse von Daten aus unterschiedlichen Quellen und Bereichen zu der Beantwortung von neuen wissenschaftlichen Fragestellungen führen könnte. Speziell in der Krebsforschung wurden bereits einige Versuche unternommen, eine Infrastruktur für eine internationale Kooperation aufzubauen. Auf einige dieser Projekte soll im Folgenden kurz eingegangen werden (Übersicht aus [19]).

- cancer Biomedical Informatics Grid (caBIG)

Das vom US National Cancer Institute's (NCI) (Rockville, USA) entwickelte cancer Biomedical Informatics Grid (caBIG) war ein Open Source Informationsnetzwerk zum Datenaustausch von klinischen Studiendaten für die Krebsforschung. Des Weiteren entwickelte caBIG unterschiedliche Tools, die eine Verknüpfung von Daten ermöglichen sollten. Die caTRIP-Plattform wurde 2006 entwickelt, um Abfragen über Patienten mit ähnlichen Profilen tätigen und deren Daten analysieren zu können. Für die Verknüpfung von Daten wurde auf die Technologie des Grid Computing gesetzt [20]. Im Jahr 2012 wurde das Projekt durch das Nachfolgeprojekt National Cancer Informatics Program (NCIP) abgelöst [21].

- cBio Cancer Genomics Portal

Das 2012 eingeführte cBio Cancer Genomics Portal ist ein Open Source Portal zur Integration von genomischen Daten. Ziel ist es, Daten aus Projekten wie dem Cancer Genome Atlas [22] und dem International Cancer Genome Consortium [23] für die Forschung verfügbar zu machen. Auch de-identifizierte klinische Daten, wie z.B. die Überlebensrate und Phänotyp-Beschreibung, sind inkludiert [24].

- Georgetown Database of Cancer (G-DOC)

Die Georgetown Database of Cancer ist eine Plattform für die Integration von

klinischen und biologischen Daten. Die Georgetown Clinical and Omics Development Engine (G-CODE) beinhaltet eine Reihe von Tools für die Datenanalyse und Visualisierung [25].

- **Integrated Clinical Omics Database (iCOD)**
Die Integrated Clinical Omics Database wurde entwickelt, um klinisch-pathologische Daten mit molekularen Daten zu verknüpfen und zur Verfügung zu stellen. Der Fokus liegt dabei auf Patienten mit hepatozellulärem Karzinom, Dickdarmkrebs und Mundhöhlenkarzinom [26].
- **Integrating Data for Analysis, Anonymization and Sharing (iDASH)**
iDash bietet Tools und Algorithmen für eine gemeinsame Nutzung und Analyse von Daten unter Berücksichtigung von Privatsphäre-Aspekten entsprechend dem Health Insurance Portability and Accountability Act of 1996 (HIPAA) [27] Konzept [28].
- **Translational Medicine Data Warehouse (tranSMART)**
tranSMART ist eine Open Source Wissensmanagement-Plattform, die Forschern eine Überprüfung ihre Hypothesen durch eine Analyse der Korrelation von genetischen und phänotypischen Daten ermöglicht [29].

Einige dieser Projekte berücksichtigen auch pädiatrische Onkologiedaten. In der G-DOC sind Daten von 256 Patienten aus 5 unterschiedlichen klinischen Studien gespeichert [25]. Jedoch fehlt bei all den genannten Lösungsansätzen die Möglichkeit einer vollständigen Integration in bereits bestehende Analysepipelines aufgrund einer mangelnden Anwendung von Datenstandards wie CDISC ODM und HL7 CDA [19].

2.2. Clinical Document Architecture

Clinical Document Architecture (CDA) ist ein von Health Level 7 (HL7) entwickelter, Extensible Markup Language (XML) basierter Datenstandard für den Austausch von klinischen Daten. Es handelt sich dabei um klinische Dokumente wie beispielsweise einen Arztbrief oder einen Befundbericht [30].

Ein CDA-Dokument gliedert sich in folgende Elemente:

- CDA-Header (Metadaten zum Dokument sowie zum Patienten)
- CDA-Body (medizinische Daten)

Des Weiteren werden CDA-Dokumente in drei Levels, die sich in der Strukturierung ihres Inhalts unterscheiden, eingeteilt [30]:

- Level 1
Einteilung des Dokuments in einen CDA-Header und in einen unstrukturierten CDA-Body. Der CDA-Body kann beispielsweise auch ein PDF-Dokument oder eine Bilddatei sein.
- Level 2
Erweiterung von Level 1 CDA-Dokumente um syntaktische Interoperabilität durch eine Gliederung des CDA-Bodys in Sektionen. Innerhalb dieser *sections* wird ein nicht maschinell lesbarer Text, wie beispielsweise eine Diagnose, hinzugefügt.
- Level 3
Ermöglicht eine maschinelle Verarbeitung von CDA-Dokumenten. Durch die Verwendung standardisierter Codes wie Logical Observation Identifiers Names and Codes (LOINC) und Systematisierte Nomenklatur der Medizin (SNOMED) wird semantische Interoperabilität erreicht. Es erfolgt eine Unterteilung der *sections* in weitere Unterelemente. Es handelt sich hierbei z.B. um das Element *entry*, das zusätzlich zum einfachen Text der *sections*, maschinell lesbare Information enthält. Es könnte sich hier beispielsweise um die International Statistical Classification of Diseases and Related Health Problems (ICD) codierte Version der Diagnose handeln.

In Abbildung 2.1 ist ein Auszug der Elemente eines CDA-Dokuments grafisch dargestellt. Das Element *templateID* ist für die Versionierung der CDA-Dokumente relevant. Informationen über den Patienten werden im Element *recordTarget* aufgelistet. Das Element *observation* beinhaltet allgemeine oder spezifische Beobachtungen, wie z. B. Diagnosen, Befunde oder Laborergebnisse [31].

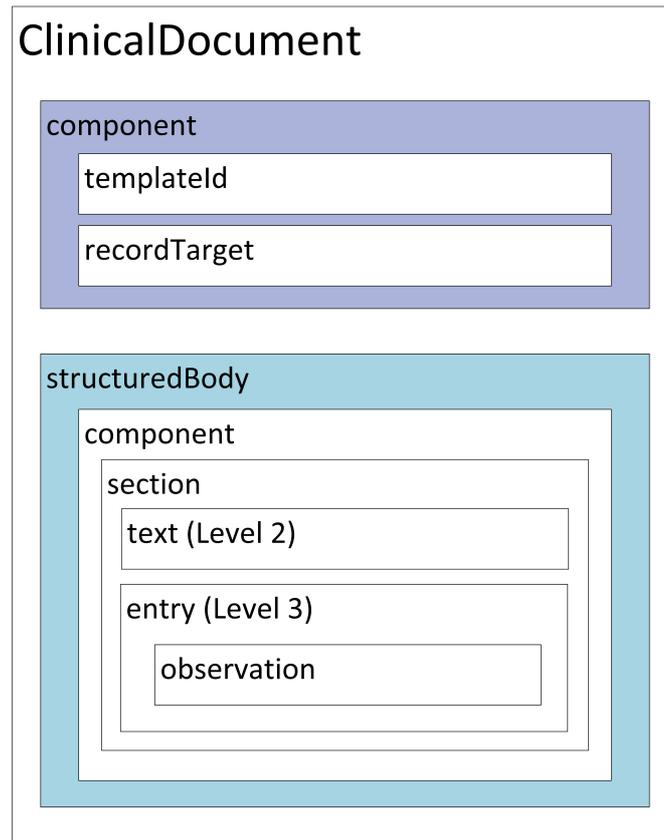


Abbildung 2.1.: Grafische Darstellung einiger Clinical Document Architecture (CDA) Elemente. Level 1 CDA-Dokumente besitzen einen unstrukturierten CDA-Body. Bei Level 2 CDA-Dokumenten erfolgt eine Gliederung des CDA-Bodys in *sections*. Level 3 CDA-Dokumente enthalten im Element *entry* zusätzlich maschinell lesbare Informationen.

2.3. Operational Data Model

Als einheitlicher Datenstandard wurde in ENCCA das von Clinical Data Interchange Standards Consortium (CDISC) zur Verfügung gestellte Operational Data Model (ODM) gewählt. ODM eignet sich besonders für die Forschung, da der XML-basierte Standard auf den Austausch und die Archivierung von klinischen Studiendaten ausgelegt ist.

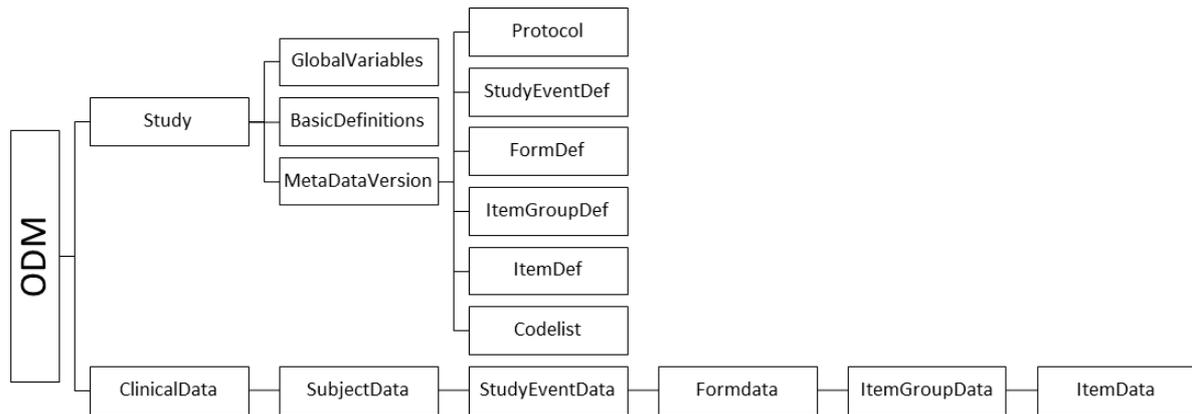


Abbildung 2.2.: Grafische Darstellung der Ebenen des Operational Data Model (ODM)

Das ODM basiert auf einer Einteilung der klinische Daten in Entitäten. In Abbildung 2.2 ist der Aufbau eines CDISC ODM-Dokuments grafisch dargestellt. In der ersten Ebene erfolgt eine Gliederung in die Hauptelemente:

- *Study* (Metadaten)
- *ClinicalData* (Klinische Daten)

Diese teilen sich ihrerseits jeweils wieder in weitere Unterelemente auf. Details für beide Hauptelemente werden im Folgenden näher beschrieben.

2.3.1. ODM-Element Study

Im ODM-Element *Study* sind Informationen über den Aufbau der Studie enthalten.

- *GlobalVariables*: Studienname, Beschreibung, Protokollname
- *BasicDefinitions*: Definition von Messgrößen
- *MetaDataVersion* : Definition der Struktur

In *MetaDataVersion* sind Informationen über die für die Studie relevanten Formulare enthalten [32]. Eine Studie besteht dabei aus einer Reihe von *StudyEvents*, *Forms*, *ItemGroups* und *Items*. Die einzelnen Elemente werden in Tabelle 2.1 genauer beschrieben.

Tabelle 2.1.: Subelemente von MetaDataVersion [32]

Tag	Kindelement	Beschreibung
<i>Protocol</i>	<i>StudyEventRef</i>	In <i>Protocol</i> sind die <i>StudyEvents</i> der Studie aufgelistet. Eine Studie kann sowohl mehrere <i>Protocols</i> als auch mehrere <i>StudyEvents</i> besitzen.
<i>StudyEventDef</i>	<i>FormRef</i>	Ein <i>StudyEvent</i> entspricht dem Auftreten eines Ereignisses, wie zum Beispiel die Registrierung eines Patienten in einem Studiensystem oder die Visite eines Arztes.
<i>FormDef</i>	<i>ItemGroupRef</i>	In <i>FormDef</i> sind die Formulare, die dem <i>StudyEvent</i> zugeordnet sind, definiert.
<i>ItemGroupDef</i>	<i>ItemRef</i>	Eine <i>ItemGroup</i> ist ein Set von zusammengehörigen <i>Items</i> . <i>ItemGroups</i> können als Abschnitte innerhalb eines Formulars betrachtet werden.
<i>ItemDef</i>	<i>CodeListRef</i>	Ein <i>Item</i> entspricht einem auszufüllenden Feld in einem Formular.
<i>CodeList</i>	<i>EnumeratedItem</i>	Definiert die möglichen Werte, die ein <i>Item</i> annehmen kann.

Jedes dieser Elemente besitzt ein Attribut, das einen eindeutigen *Object Identifier (OID)* enthält [32]. Die *OIDs* werden verwendet um Elemente innerhalb des Dokuments eindeutig zu kennzeichnen und Verbindungen zwischen ihnen darzustellen. So referenzieren die in *Protocol* enthaltenen *StudyEventRefs* über die *OID* auf die entsprechenden *StudyEventDefs* (siehe Abbildung 2.3).

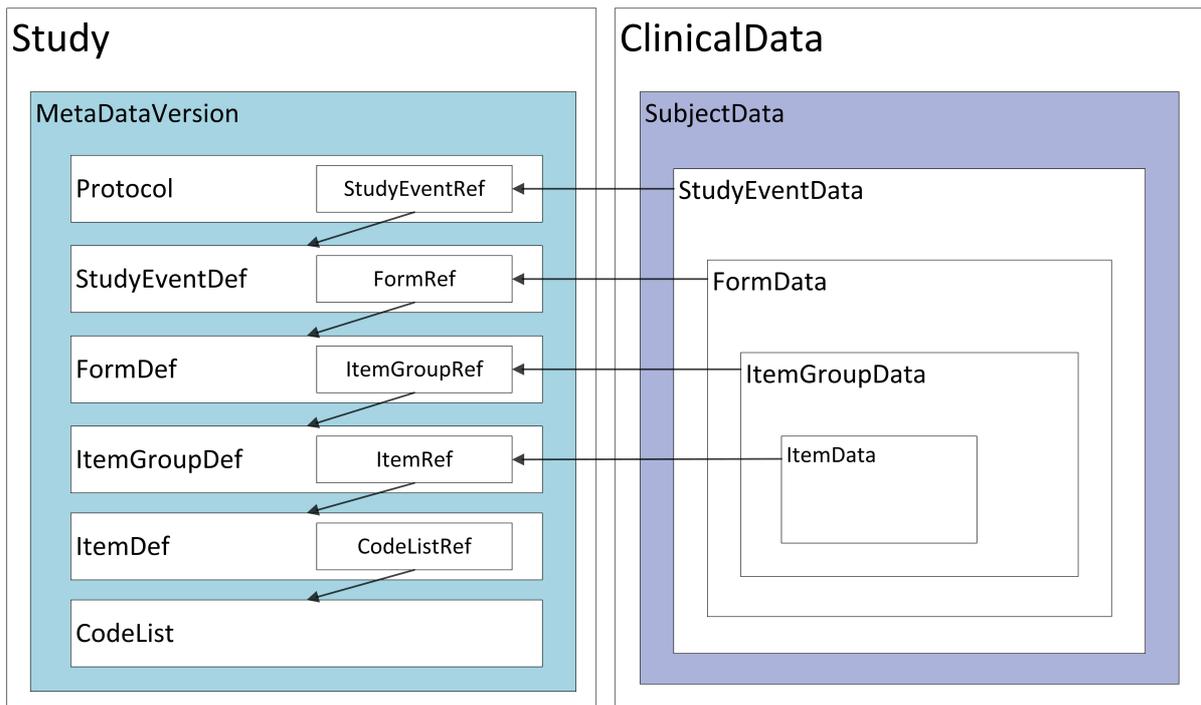


Abbildung 2.3.: Grafische Darstellung der Verbindung der unterschiedlichen Elemente im Operational Data Model (ODM) mittels *Object Identifiers (OIDs)*.

Attribute

Für die unterschiedlichen Elemente können noch weitere Attribute zur genaueren Definition hinzugefügt werden.

Das ODM-Element *Protocol* beinhaltet die Liste der *StudyEventRefs*. Das Attribut *OrderNumber* ist dabei lediglich eine nummerierte Reihenfolge der *StudyEventDefs*, für den Fall, dass Daten in irgendeiner Form präsentiert werden [32]. Ist ein Tag als *Mandatory* gekennzeichnet, so ist diese Instanz obligatorisch in der Studie enthalten.

Die Attribute *SASDatasetName* und *SASFieldName* stellen einen maximal 8 bzw. 32 Zeichen langen String dar. Dieser ist für eine statistische Auswertung mit den von SAS Institute (Cary, USA) entwickelten Softwarelösungen von Bedeutung [33].

Ist die Eigenschaft *Repeating* als *Yes* gekennzeichnet, kann ein Element mit seiner *OID* mehrmals vorkommen. Dies ist der Fall, wenn es sich um möglicherweise wiederkehrende *StudyEvents*, wie zum Beispiel dem Auftreten von Nebenwirkungen, handelt.

Zusätzlich zur Unterscheidung von wiederkehrenden und nicht wiederkehrenden *StudyEvents*, erfolgt eine Einteilung in drei unterschiedliche Typen [32]:

- *Scheduled*
Ein *StudyEvent* wird als *Scheduled* bezeichnet, wenn es sich um ein erwartetes Ereignis, wie beispielsweise eine geplante Visite, handelt.
- *Unscheduled*
Außerplanmäßige *StudyEvents* können auftreten, müssen aber nicht. Ein Beispiel wäre eine Reihe von Formularen, die für ein vorzeitiges Beenden einer Chemotherapie, aufgrund eines schwerwiegenden unerwünschten Ereignis, benötigt werden.
- *Common*
Dazu zählen *StudyEvents*, die an mehreren unterschiedlichen Datenerhebungen beteiligt sind, wie zum Beispiel unerwünschte Ereignisse oder Begleitmedikation.

Im Listing 2.1 ist ein Beispiel des Elements *MetaDataVersion* anhand eines Registrierungsformulars dargestellt. Da eine Registrierung nur einmalig durchgeführt wird, handelt sich um ein nicht wiederkehrendes *StudyEvent* und die Eigenschaft *Repeating* ist somit auf *No* gesetzt. Des Weiteren handelt es sich um ein ungeplantes *StudyEvent*, weswegen der *Type* als *Unscheduled* definiert wird.

Listing 2.1: Beispiel des Elements *MetaDataVersion*, anhand eines Registrierungsformulars.

```
<MetaDataVersion OID="MD.1" Name="1">
  <Protocol>
    <StudyEventRef StudyEventOID="SE.register" Mandatory="Yes"/>
  </Protocol>
  <StudyEventDef OID="SE.register" Name="Registration" Type="Unscheduled" Repeating="No">
    <FormRef FormOID="FORM.register" Mandatory="Yes"/>
  </StudyEventDef>
  <FormDef OID="FORM.register" Repeating="No" Name="Registration Form">
    <ItemGroupRef ItemGroupOID="IG.personal_data" Mandatory="Yes"/>
  </FormDef>
  <ItemGroupDef OID="IG.personal_data" Name="Personal Data" SASDatasetName="persdat">
    <ItemRef ItemOID="Age" OrderNumber="1" Mandatory="Yes"/>
    <ItemRef ItemOID="Sex" OrderNumber="2" Mandatory="Yes"/>
  </ItemGroupDef>
  <ItemDef OID="ID.Age" Name="Age" DataType="integer" SASFieldName="age"/>
  <ItemDef OID="ID.Sex" Name="Sex" DataType="text" SASFieldName="sex">
    <CodeListRef CodeListOID="CL.Sex"/>
  </ItemDef>
  <CodeList OID="CL.Sex" DataType="text">
    <EnumeratedItem CodedValue="Male"/>
    <EnumeratedItem CodedValue="Female"/>
  </CodeList>
</MetaDataVersion>
```

2.3.2. ODM-Element *ClinicalData*

Das ODM-Element *Study* enthält also die Metadaten einer Studie und legt damit fest, welche Daten in der Studie aufgezeichnet werden können. Die tatsächlichen, klinischen Daten der Patienten befinden sich im ODM-Element *ClinicalData* (Abbildung 2.2). Für jeden Patienten wird ein Element *SubjectData* angelegt und die *StudyEvents*, *Forms*, *ItemGroups* und *Items* aufgelistet, für die Werte vorhanden sind. Die Patienten-ID ist im Attribut *SubjectKey* gespeichert. *StudyEventData*, *FormData*, *ItemGroupData* und *ItemData* beziehen sich dabei auf die in *MetaDataVersion* definierten Elemente über die entsprechende *OID* (siehe Abbildung 2.3).

Für den Fall, dass es sich um ein wiederkehrendes *StudyEvent* handelt, wird für jedes Auftreten des *StudyEvents* in dem Element *ClinicalData* ein neues Element *StudyEventData* erzeugt und mit einem *StudyEventRepeatKey* versehen. Durch diesen Schlüssel ist es möglich wiederkehrende *StudyEvents* voneinander zu unterscheiden. Für die Elemente *Form* und *ItemGroup* wird eine entsprechende Unterscheidung der wiederkehrenden *Forms* und *ItemGroups* durch die Schlüssel *FormRepeatKey* und *ItemGroupRepeatKey* ermöglicht.

Die tatsächlichen Werte sind dem Tag *ItemData* als Attribut hinzugefügt. Im Attribut *StudyOID* ist eine *OID* zur eindeutigen Identifizierung der Studie enthalten.

Listing 2.2: Beispiel des Elementes *ClinicalData* eines nicht wiederkehrenden *StudyEvents*, anhand der im Listing 2.1 definierten Metadaten.

```
<ClinicalData StudyOID="ST.1" MetaDataVersionOID="MD.1">
  <SubjectData SubjectKey="Patient 1">
    <StudyEventData StudyEventOID="SE.register">
      <FormData FormOID="FORM.register">
        <ItemGroupData ItemGroupOID="IG.personal_data">
          <ItemData ItemOID="ID.Age" Value="3"/>
          <ItemData ItemOID="ID.Sex" Value="Male"/>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
```

2.3.3. Konformität

Eine XML-Datei entspricht nur dann dem ODM-Standard, wenn alle Aspekte der Norm berücksichtigt werden. Diese Kriterien umfassen sowohl syntaktische als auch semanti-

sche Einschränkungen [32]. Eine ODM-konforme Datei ...

1. ... muss eine valide XML-Datei sein.
2. ... muss dem XML-Namespace-Standard genügen.
3. ... darf nur Elemente beinhalten, die im ODM-Standardschema enthalten sind und den Verschachtelungsregeln folgen.
4. ... muss einen Prolog und an oberste Ebene das Element *ODM* enthalten (siehe Listing 2.3).
5. ... muss den Namespace ODM 1.3 enthalten.
6. ... muss ein Attribut *ODMVersion* in obersten Ebene mit der Versionsnummer enthalten

Listing 2.3: Beispiel des Prologs und der obersten *ODM*-Ebene der ODM-Version 1.3.2 [32].

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.3 ODM1-3-2.xsd"

  ODMVersion="1.3.2"
  FileOID="000-00-0000"
  FileType="Snapshot"
  Description="Sample ItemData Extension"

  AsOfDateTime="2013-02-04T07:57:00">
```

2.4. CDISC ODM Checker Tool

Um ODM-XML-Dokumente auf ihre ODM-Konformität überprüfen zu können, wurde von XML4Pharma (Thal, Österreich) das sogenannte CDISC ODM Checker Tool entwickelt [34]. Das Tool ist in drei Versionen erhältlich:

- Base Edition
- Professional Edition
- Enterprise Edition

Für diese Arbeit wurde die Base Edition verwendet. Diese wird für CDISC-Members kostenlos zur Verfügung gestellt. Ein Screenshot des Systems ist in Abb. 2.4 dargestellt.

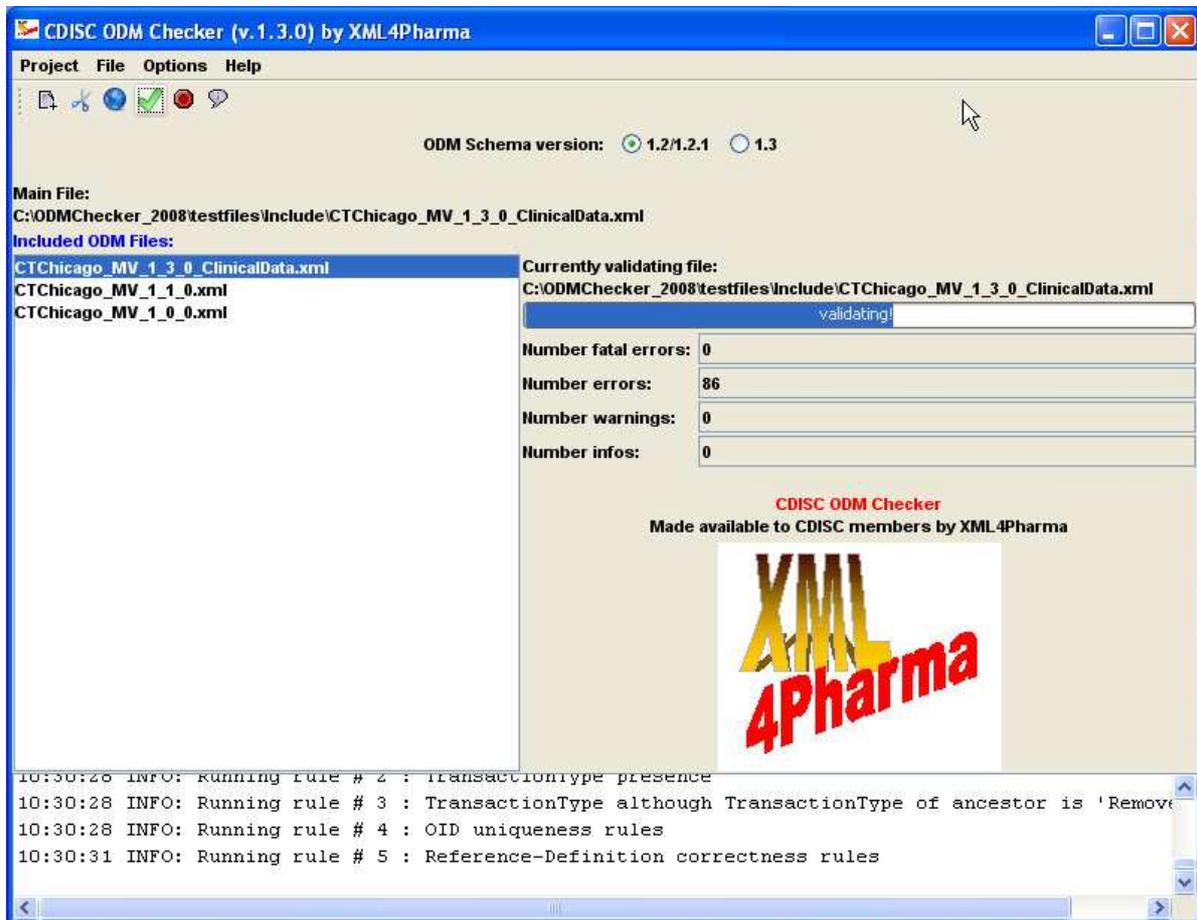


Abbildung 2.4.: Screenshot des CDISC ODM Checker Tools von XML4Pharma (Thal, Österreich) [34].

2.5. LTI-Studiensystem

Die LTI-Studie läuft seit November 2012 und ist in Zentren in Österreich, Australien, Frankreich, Deutschland, Irland, Israel, Italien, Spanien und Großbritannien verfügbar. Derzeit sind 124 Patienten im Studiensystem registriert (Stand 2014-09-29).

In die LTI-Studie werden Patienten mit refraktären (dh. sie reagieren nicht auf die initiale Behandlung) oder rezidivierten (d.h. ein zuvor behandelter Tumor ist erneut aufgetreten) Neuroblastom eingeschlossen [12].

Die Wirkung folgender Medikamente wird in der Studie untersucht [35]:

- ch14.18/CHO (monoklonaler Antikörper für die Immuntherapie)
- Aldesleukin (Interleukin für die biologische Therapie)
- Isotretinoin (Standardtherapie bei Neuroblastompatienten)

Von der Verabreichung von ch14.18/CHO wird eine Verbesserung des Outcomes von Patienten mit Neuroblastom erwartet. Jedoch sind starke Schmerzen als Nebenwirkung für dieses Medikament bekannt, wodurch oft hohe Dosen von intravenösen Morphinen erforderlich sind. In anderen klinischen Studien wurde gezeigt, dass eine kontinuierliche aber dadurch geringer dosierte Verabreichung des Medikaments über eine längere Zeit (z.B. 10 Tage) bei identer Gesamtdosis eine Reduktion der Nebenwirkung bewirkt [35].

Hauptziel der Studie ist es, einen Weg zu finden, wie ch14.18/CHO in Kombination mit subkutanem Aldesleukin und Isotretinoin am wirkungsvollsten eingesetzt werden kann.

Das LTI-Studiensystem ist das System zur elektronischen Datenerfassung. Das System ist mit Zope 2.10 (Zope Foundation Inc, Richardson, USA) in Python 2.4 (Python Software Foundation, Delaware, USA) [36] realisiert und läuft auf einer virtualisierten Server-Infrastruktur von AIT. Alle Daten werden in einer PostgreSQL-Datenbank (PostgreSQL Global Development Group, San Francisco, USA) gespeichert. Der Zugriff auf das System für Ärzte und andere Studien-Mitarbeiter erfolgt web-basiert über sichere HTTPS-Verbindungen. Insgesamt sind im Studiensystem 33 Formulare zur Eingabe von Studiendaten zur LTI-Studie abgebildet. Alle Patientendaten werden in pseudonymisierter Form abgelegt.

Ein Screenshot des LTI-Studiensystems ist in Abbildung 2.5 dargestellt.

The screenshot shows a web application interface for patient registration. At the top, a navigation bar contains links: Start, Queries and SDV, Patient list, Registry (highlighted), Administration, Overall Progress, Adverse Events, Messages, Statistics, and Downloads. Below the navigation bar, a header indicates 'Step 1 of 1' with a person icon. The main content area is titled 'Registry' and contains a form with the following fields:

- Register patient** (button)
- *Hospital where patient is registered**: A dropdown menu with 'Please select'.
- *SIOPEN-R-NET patient ID**: A text input field.
- *Date of birth**: A date picker with a calendar icon and the format 'YYYY-mm-dd'.
- *Gender**: Radio buttons for 'Male' and 'Female'.
- *Ethnic origin**: A dropdown menu with 'Please select'.

Fields marked with an asterisk (*) are mandatory. At the bottom of the form, there is another 'Register patient' button and a 'Top of page' link.

Abbildung 2.5.: Screenshot der Registrierungsfunktion des LTI-Studiensystems

2.6. ABCD-4-E

Die Advanced Biomedical Collaboration Domain for ENCCA (ABCD-4-E) ist eine Kommunikationsplattform, die aktuell für die Zusammenarbeit der ENCCA-Partner entwickelt wird (siehe Abbildung 2.6).

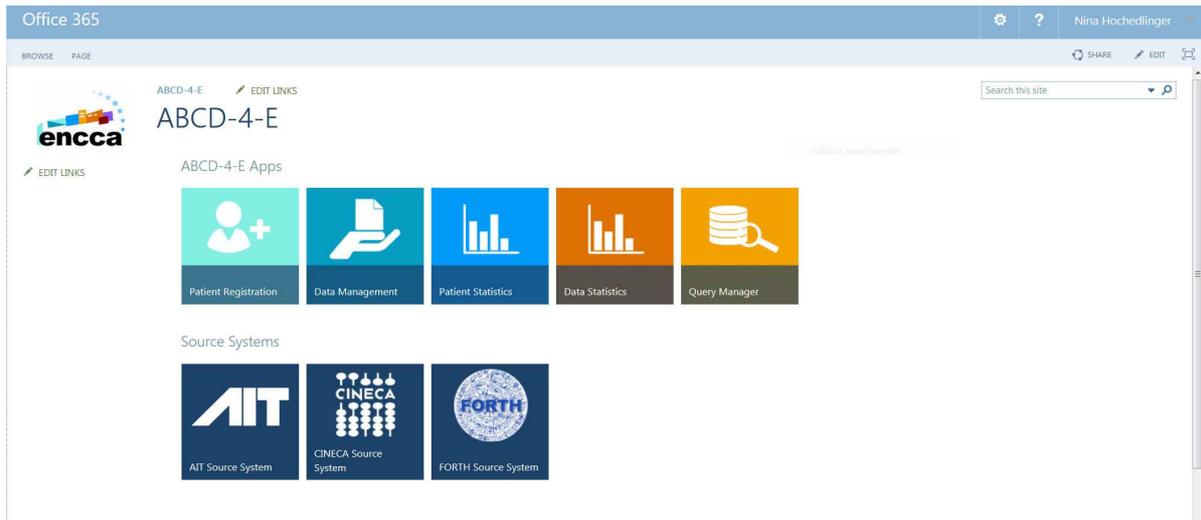


Abbildung 2.6.: Screenshot des Advanced Biomedical Collaboration Domain for ENCCA (ABCD-4-E) Portals, in dem unterschiedliche Web-Applikationen, die eine Kooperation zwischen den ENCCA-Mitgliedern ermöglichen sollen, bereitgestellt werden.

In Abbildung 2.7 ist das Konzept des Prozesses von den *Dataset Sources* über das ABCD-4-E-Portal bis hin zum *Dataset Consumer*, dargestellt. *Dataset Sources* können beispielsweise klinische Studiensysteme oder Biobanken sein.

Im ABCD-4-E-Portal werden verschiedene Web-Applikationen, die eine Kooperation zwischen Mitgliedern ermöglichen sollen, bereitgestellt. Unter anderem soll die Möglichkeit der horizontalen und vertikalen Verknüpfung von Daten aus unterschiedlichen *Dataset Sources* für eine sekundäre Forschung realisiert werden.

Interoperabilität spielt dabei eine zentrale Rolle. Es wurden daher bestimmte Voraussetzungen für die *Dataset Sources* definiert. So müssen alle beteiligten *Dataset Sources* die Möglichkeit eines ODM-Exports haben. Für die unterschiedlichen *Dataset Sources* soll in Zukunft ein einheitliches Pseudonymisierungs-Konzept verwendet werden. Die Generierung der Pseudonyme soll dabei mit der sogenannten Patient Registration App erfolgen.

In einer weiteren App sollen vordefinierte, für eine Sekundärforschung relevante, Kerndatensätze aus den Quelldaten extrahiert werden. Die Quelldokumente müssen dabei im ODM-Format verfügbar sein. Im ABCD-4-E-Portal sollen dann die ODM-Dateien aus den unterschiedlichen Quellen in ein *Repository* hochgeladen werden. Dabei sollen in

einer *Registry* Metadaten über die Quelldokumente in einer Metadatenbank gespeichert werden.

Auf der Ebene des *Dataset Consumers* sollen diese Dokumente dann mit einer weiteren App, die als *QueryManager* fungiert, unter der Berücksichtigung bestimmter Berechtigungskriterien, abgeholt werden können. Die Dokumente können dann für eine statistische Analyse herangezogen werden.

Die beiden Tools, die im Zuge dieser Arbeit implementiert werden sollen, stellen wichtige Schritte in diesem Prozess dar. Durch die Entwicklung des Tools zum Exportieren von klinischen Studiendaten aus der LTI-Studie in ein standardisiertes Datenformat soll syntaktische Interoperabilität gewährleistet werden. Der ODM-Export setzt auf der Ebene der *Dataset Sources* an.

Das Tool zum Extrahieren von vordefinierten Kerndatensätzen soll als Web-App in das ABCD-4-E-Portal eingebunden werden und soll die Dokumente aus den unterschiedlichen Quellsystemen für eine Analyse der Daten hinsichtlich semantischer Interoperabilität vorbereiten.

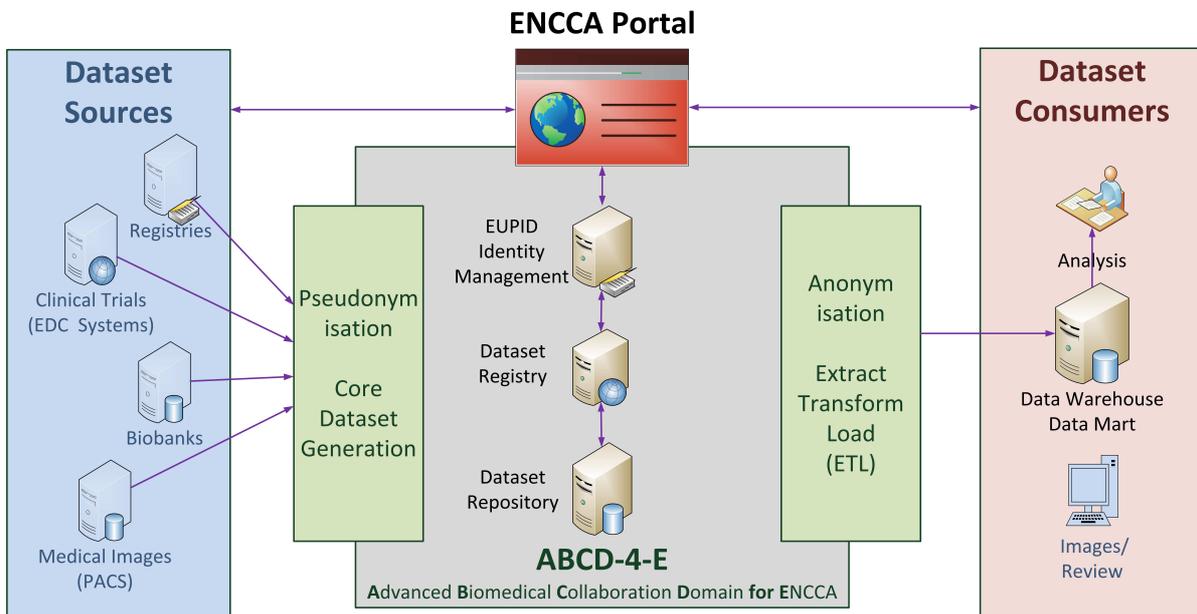


Abbildung 2.7.: Konzept der Eingliederung des ABCD-4-E-Portals in die Gesamtarchitektur.

Alle Apps in der ABCD-4-E-Plattform sollen mit Bootstrap [37] realisiert werden. Es

handelt sich um SharePoint hosted Apps [38, 39]. Auf zwei der für die vorliegende Arbeit wesentlichen Apps wird im Folgenden näher eingegangen.

2.6.1. Patient Registration App und Pseudonymisierung

Der Schutz der Patientendaten hat im Projekt ENCCA einen hohen Stellenwert. Um ein entsprechendes Maß an Datensicherheit gewährleisten zu können, wurde im Vorfeld ein Konzept entwickelt, das sowohl den Zielen von ENCCA, als auch den regulatorischen Anforderungen des Datenschutzes gerecht wird [18].

Teil des Konzeptes ist die Pseudonymisierung von Daten. Dabei wird der Name des Patienten durch einen Identifikator ersetzt, der es nur autorisierten Personen ermöglicht, auf die Identität des Patienten zurückzuschließen. [40].

Für die Generierung der Pseudonyme wird die Patient Registration App verwendet. Nimmt ein Patient in mehreren klinischen Studien teil, so wird für jeden Kontext ein spezifisches Pseudonym generiert (siehe Abbildung 2.8). Benutzer und Studienärzte einer Studie A (z.B. LINES) können so über die Pseudonyme, die ihrem Studienkontext zugeordnet sind, Daten einem Patienten aus ihrer Studie zuordnen, Studiendaten desselben Patienten aus einem Studienkontext B können diesem Patienten jedoch nicht zugeordnet werden.

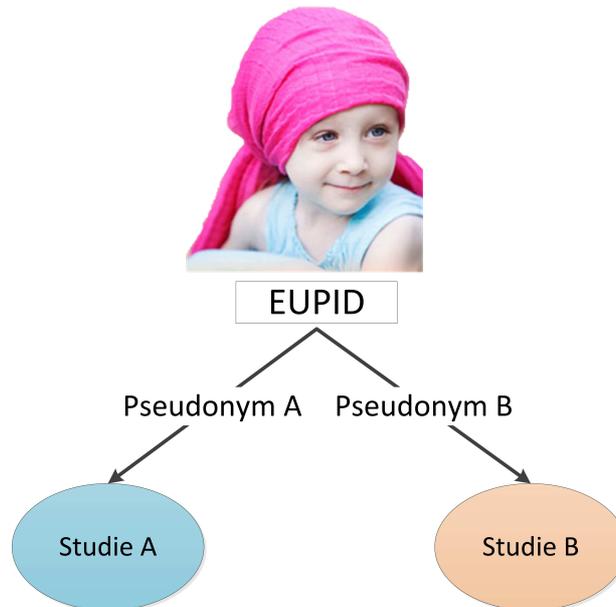


Abbildung 2.8.: Kontextspezifische Pseudonymisierung nach dem ENCCA Unified Patient Identifier (EUPID) Konzept. Studienärzte und Benutzer der Studie A können den Patienten keine Daten aus Studie B zuordnen.

Um eine horizontale und vertikale Integration ermöglichen zu können, werden die Pseudonyme aus unterschiedlichen Studien (Studie A und Studie B), die zu einem Patienten gehören, im Hintergrund über die EUPID miteinander verknüpft und verschlüsselt. Nur eine Trusted Third Party, die in Besitz eines privaten Schlüssels ist, könnte eine Re-Identifikation der Daten auf Anforderung durchführen. Dies könnte beispielsweise bei Vorliegen eines entsprechenden Ethikkommissionsvotums und Einverständniserklärungen der Patienten möglich sein. Durch die Verknüpfung der Daten könnten dadurch im Zuge der sekundären Forschung Daten aus Studie A mit Daten aus Studie B verknüpft werden [18].

2.6.2. Data Management App

Die vorliegende Arbeit stellt einen wesentlichen Teil der Data Management App des ABCD-4-E-Portals dar. Im Folgenden wird daher auf die Data Management App und die Einbindung der im Zuge der Arbeit entwickelten Tools in die App im Detail eingegangen.

Mit der Data Management App soll die horizontale und vertikale Integration von Daten aus unterschiedlichen Datenquellen ermöglicht werden. Dazu werden die von ENCCA vordefinierte Kerndatensätze aus den Quelldokumenten extrahiert. Dokumente, die mit

der Data Management App transformiert werden sollen, müssen gültige ODM-XML-Dokumente sein.

Im ersten Schritt wird das ODM-Quelldokument durch den User ausgewählt. ODM-Quelldokumente müssen zuvor aus Quellsystemen (z.B. aus einem Studiensystem) exportiert worden sein. Danach können Patienten, für die die Extraktion der Kerndatensätze durchgeführt werden soll, ausgewählt werden. Die Data Management App überprüft, aufgrund der Struktur des ODM-Elements *SubjectKey*, ob ein mit der Patient Registration App generiertes EUPID-Pseudonym vorliegt. In Abbildung 2.9 wird beispielsweise beim ersten Patienten das Pseudonym *LTI1* als nicht EUPID-konform identifiziert. Der User könnte in diesem Fall nachträglich ein EUPID-Pseudonym generieren und dieses in dem Textfeld *Set EUPID* angeben.

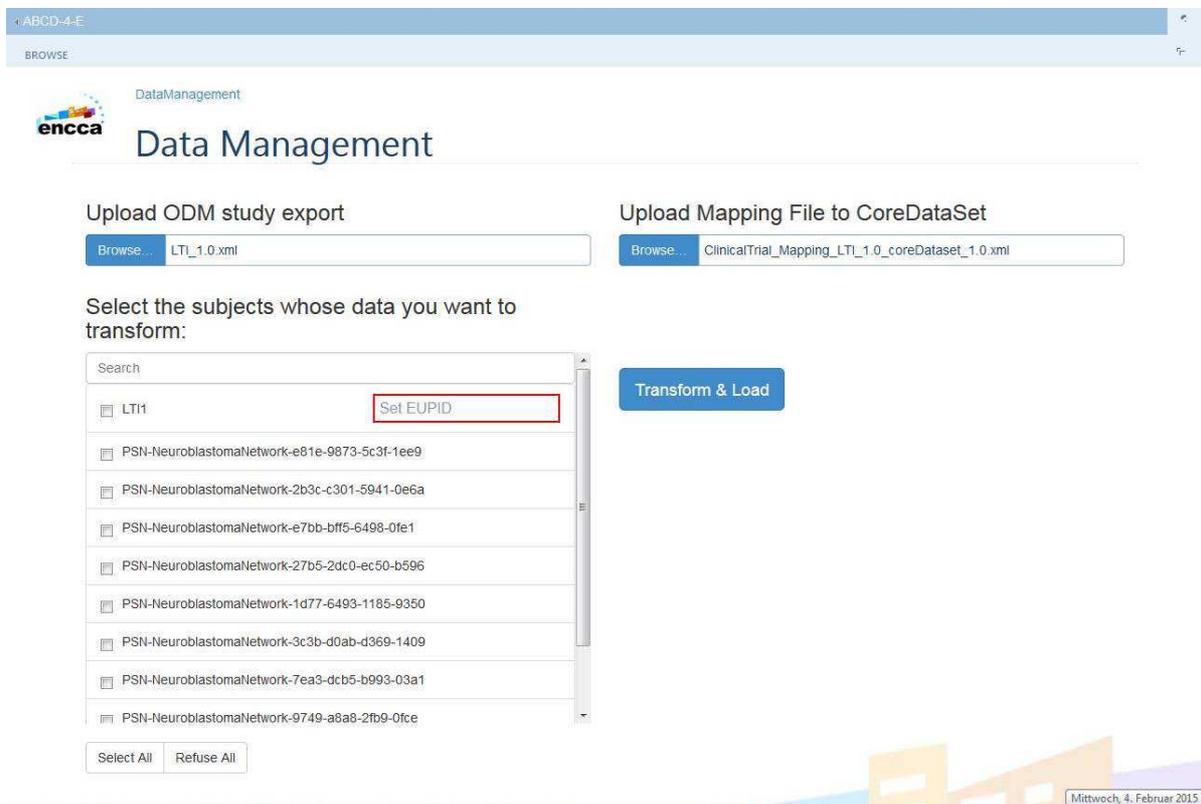


Abbildung 2.9.: Data Management App: Upload der LTI-ODM-Exportdatei. Nachträglich generierte EUPID-Pseudonyme können im Textfeld *Set EUPID* angegeben werden.

Durch Klick auf den Button *Transform & Load* wird die ODM-Transformation gestartet. Diese Funktionalität wurde im Zuge dieser Arbeit entwickelt und wird im Kapitel 3.2 beschrieben.

2.7. ENCCA-Kerndatensatz

Ein Kerndatensatz beinhaltet Daten, die aus den Quelldokumenten extrahiert werden sollen. In ENCCA sind bisher zwei unterschiedliche Typen von Kerndatensätzen definiert:

- Kerndatensatz für klinische Studien (KDKS)
- Kerndatensatz für Bioproben (KDBP)

In einem KDKS sind jene Daten enthalten, die aus den klinischen Studien extrahiert werden sollen. Beispiele dafür sind: Alter, Geschlecht, Tumorklassifikation nach International Classification of Diseases for Oncology (ICD-O-3) [41] etc. (siehe Tabelle 2.2).

Der KDBP ist als Teil der Minimum Data Set for Sharing Biobank Samples, Information, and Data (MIABIS) definiert [42].

Beide Kerndatensatz-Definitionen werden von ENCCA-Partnern als Excel-Datei zur Verfügung gestellt. Zum einen werden die Daten aufgelistet, die aus den Dateien extrahiert werden. Zum anderen beinhaltet der Kerndatensatz auch eine Definition der Werte, mit denen die Daten abgebildet werden sollen. So soll zum Beispiel *Sex at birth* statt den Werten *Male* und *Female* die Werte *1* und *2* enthalten. Dadurch sollen die aus unterschiedlichen Quellen stammenden Daten standardisiert werden um eine anschließende Auswertung der Daten durchführen zu können.

Zukünftige Studiensysteme sollen diese Kerndatensätze als Basis verwenden und -abhängig von der jeweiligen Fragestellung - um Daten, die speziell für die einzelne Studie erfasst werden sollen, erweitern.

Tabelle 2.2.: Ein Auszug des in ENCCA definierten Kerndatensatz für klinische Studien (KDKS). Diese Daten sollen aus den Exportdateien extrahiert werden. Der KDKS beinhaltet auch eine Definition der Werte, mit denen die Daten abgebildet werden sollen.

Variable	Values	Definition
Ethnic group		
Patient identification number		A unique identifier of the patient in the originating registry, which is not necessarily the official personal identification number.
Date of birth		If date of birth cannot be provided in full, the next possible/available detail should be included, such as YYYYMM or YYYYDD
Sex at birth	1 = Male	
	2 = Female	
Date of registration		Date when first recorded in the registry
Age in years		Latest completed year of age at the time of diagnosis
ICD-O-3 Topography		ICD-O-3, Example: C531, 531
ICD-O-3 Morphology		ICD-O-3, Example: 8170
ICD-O-3 Behaviour		ICD-O-3, Example: 3
ICD-O-3 Basis of diagnosis	0 = DCO	ICD-O-3
	1 = Clinical	
	7 = Microscopic	
Stage	1 = Stage 1	INSS-Stage
	2 = Stage 2a	
	3 = Stage 2b	
	4 = Stage 3	
	5 = Stage 4	
	6 = Stage 4s	
	9 = unknown	

2.8. Extensible Stylesheet Language Transformations

Extensible Stylesheet Language Transformations (XSLT) wird verwendet, um aus den ODM-XML-Quelldateien die definierten Kerndatensätze zu extrahieren.

XSLT ist eine Sprache zur Transformation von XML-Dokumenten in Formate wie HTML, Text oder andere XML-Strukturen [43]. Das Kernelement der Transformation ist das XSLT-Stylesheet. Dort werden Umwandlungsregeln definiert, die das Resultat der Transformation bestimmen. Das Stylesheet besteht aus einer Reihe von Anweisungen in Kombination mit XML Path Language (XPath) Ausdrücken, die definieren, wie mit den Elementen des Quelldokuments zu verfahren ist. XPath wird verwendet, um bestimmte Elemente im Quelldokument auszuwählen.

Die Transformation wird von einem XSLT-Prozessor durchgeführt (siehe Abbildung 2.10). Der Prozessor verwendet ein oder mehrere Quelldokumente und das XSLT-Stylesheet um daraus die Ergebnisdokumente zu generieren. Die Quelldokumente bleiben dabei erhalten.

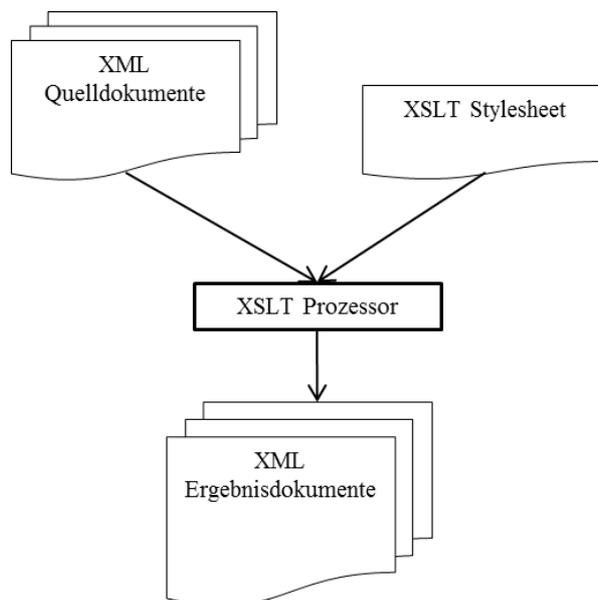


Abbildung 2.10.: XSLT Prozessor

In Tabelle 2.3 und 2.4 sind die für die Realisierung der ODM-Transformation relevanten XSLT-Funktionen und XPath-Ausdrücke beschrieben.

Tabelle 2.3.: Für die ODM-Transformation relevanten XSLT-Funktionen [43].

Funktion	Beschreibung
<code><xsl:template match="Pattern"></code>	Definition einer Templaterregel. Das Attribut <i>match</i> beinhaltet einen XPath-Ausdruck, der bestimmt, wann die Templaterregel ausgeführt wird.
<code><xsl:apply-templates/></code>	Selektiert alle Unterelemente des aktuellen Elements. Ist nur innerhalb einer <code><xsl:template match></code> Umgebung erlaubt.
<code><xsl:for-each select="Pattern"></code>	Entspricht in etwa der in Programmiersprachen verwendeten <i>for</i> -Schleife. Es werden dabei Knoten derselben Ebene gesucht.
<code><xsl:if test="Pattern"></code>	Überprüft, ob der aktuelle Knoten die angegebene Bedingung erfüllt und überspringt gegebenenfalls einen Abschnitt.
<code><xsl:choose></code> <code><xsl:when test="Pattern">...</xsl:when></code> <code><xsl:otherwise>...</xsl:otherwise></code> <code></xsl:choose></code>	Entspricht in etwa der in C definierten Switch-Case Anweisung und dient der Auswahl bestimmter Templateabschnitte.
<code><xsl:copy-of select="." /></code>	Kopiert den aktuellen Knoten inklusive aller Kindelemente.
<code><xsl:copy-of select="@*" /></code>	Kopiert alle Attribute des aktuellen Knotens.
<code><xsl:value-of select="Pattern"/></code>	Liefert den Wert entsprechend des XPath-Ausdrucks.
<code><xsl:key name="QName" match="Pattern" use="Pattern"/></code>	Schlüsselfunktion für die Auswahl von Elementen. Das Attribut <i>use</i> legt fest, auf welchen XPath-Ausdruck der Wert des Schlüssels angewendet werden soll.
<code>key('QName', 'keywert')</code>	Aufruf der Schlüsselfunktion. Gibt alle Knoten zurück, für die der genannte Schlüssel (<i>QName</i>) das übergebene Kriterium (<i>keywert</i>) erfüllt.

Tabelle 2.4.: Für die ODM-Transformation relevanten XPath-Ausdrücke [43].

Funktion	Abkürzung	Beschreibung
/		Wählt den Root-Knoten aus.
self::node()	.	Wählt den aktuellen Knoten aus.
parent::node()	..	Wählt den Eltern-Knoten des aktuellen Knotens aus.
child::node()		Wählt alle Kind-Knoten des aktuellen Knotens aus.
attribute::*	@*	Auswahl aller Attribute des aktuellen Knotens.
ancestor::		Wählt alle Vorfahren des aktuellen Knotens aus.

In Abbildung 2.11 sind die in Tabelle 2.4 beschriebenen XPath-Ausdrücke grafisch dargestellt.

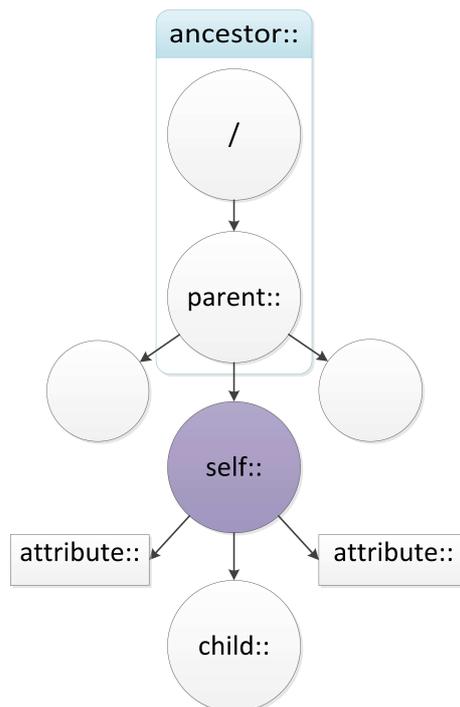


Abbildung 2.11.: Grafische Darstellung der relevanten XPath-Ausdrücke. *Self* stellt dabei den aktuellen Knoten dar. Ausgehend von diesem Knoten können die *ancestor*-, *parent*-, *attribute*- und *child*-Knoten mit XPath-Ausdrücken ausgewählt werden.

3. Methoden

Ziel der vorliegenden Arbeit war es, zwei Tools für den Datenaustausch in ENCCA zu entwickeln. Ein Tool zum Export von ODM-Daten aus dem LTI-Studiensystem, sowie ein Tool zur Transformation der exportierten ODM-Daten, sodass diese standardisiert in ein *Repository* übergeben werden können. Beide Tools werden im Folgenden näher beschrieben.

3.1. ODM-Export

Mit Hilfe des ODM-Export-Tools sollten Daten aus den relationalen Datenbanken in das standardisierte Format ODM exportiert werden. Das Tool sollte aus dem Websystem der LTI-Studie gestartet werden.

Da das LTI-Studiensystem mit dem auf Python basierenden Zope 2.10 realisiert wurde, wurde die Implementierung dieses Tools mit Python durchgeführt.

Im ODM-Format wurden grundsätzlich zwei verschiedene Arten von *StudyEvents* unterschieden:

- Nicht wiederkehrende Events (z.B. Patientenregistrierung)
- Wiederkehrende Events (z.B. Auftreten von Nebenwirkungen)

Beim ODM-Export mussten diese dementsprechend unterschiedlich behandelt werden. Beide Fälle werden im Folgenden genauer beschrieben.

3.1.1. Fall 1, nicht wiederkehrende Events

In Abbildung 3.1 ist die Zuordnung der ODM-Elemente anhand eines Auszuges des Registrierungsformulars der LTI-Studie dargestellt.

Bei der Registrierung der Patienten in der LTI-Studie wurden von den behandelnden Ärzten insgesamt acht *Items* im LTI-Studiensystem ausgefüllt. In Abbildung 3.1 werden aus Gründen der Übersichtlichkeit nur vier dieser acht *Items* beschrieben:

- das Krankenhaus, in dem der Patient registriert wurde
- das Geburtsdatum
- das Geschlecht
- die ethnische Herkunft

Im ODM-Format entsprach der Vorgang der Registrierung einem *StudyEvent*. Diesem Event wurde ein Formular (das Registrierungsformular) zugeordnet. Innerhalb des Formulars wurden alle vier *Items* derselben *ItemGroup* zugeordnet. Die einzelnen Werte für jeden Patienten wurden dem Tag *ItemData* als Attribut *Value* hinzugefügt.

Die Patienten-ID der LTI-Studie wurde in ODM nach *SubjectKey* gemappt.

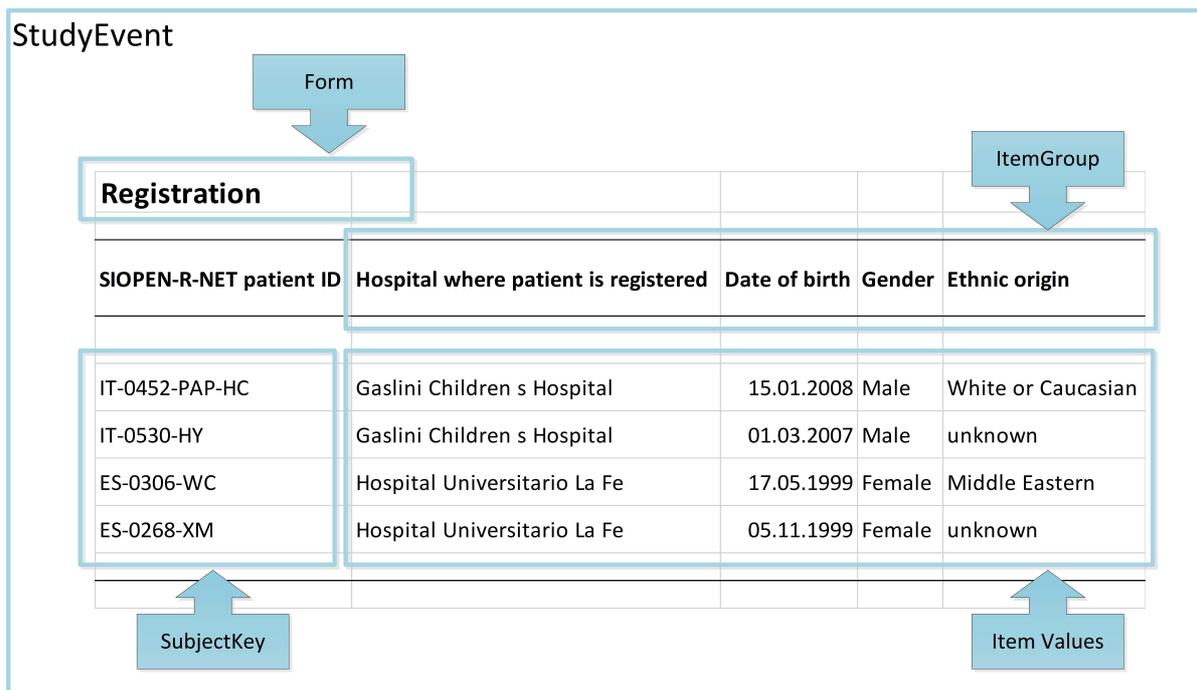


Abbildung 3.1.: Grafische Darstellung der Zuordnung der ODM-Elemente am Beispiel eines Registrierungsformulars.

3.1.2. Fall 2, wiederkehrende Events

Adverse Events sind ein typisches Beispiel für wiederkehrende Events, da diese bei Patienten öfter auftreten konnten. Beispielsweise wurde in der LTI-Studie zu jeder Nebenwirkung die Art und der Schweregrad des *Adverse Events* (z.B. *Fever*, *Pain*, *Sepsis*, *Caugh*) dokumentiert und in der Datenbank mit einer internen laufenden Nummer *Cycle* versehen (siehe Abbildung 3.2).

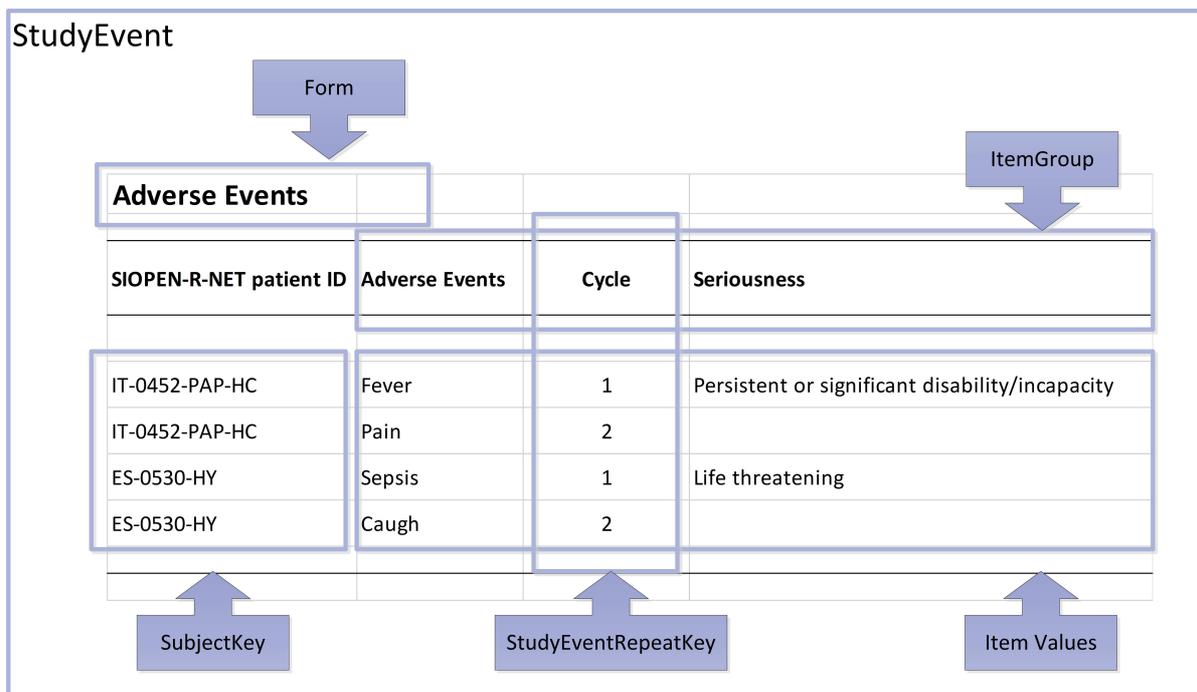


Abbildung 3.2.: Grafische Darstellung der Zuordnung der ODM-Elemente am Beispiel von *Adverse Events*.

Im ODM-Format wurde der *Cycle* über das Attribut *StudyEventRepeatKey* dargestellt. Das Vorhandensein der Spalte *Cycle* in einer Tabelle kennzeichnete Formulare, die öfter als einmal ausgefüllt werden konnten und somit beim Export unterschiedlich behandelt werden mussten. Bei wiederkehrenden *StudyEvents* wurde für jedes Auftreten ein Element *StudyEventData* angelegt und die laufende Nummer im Attribut *StudyEventRepeatKey* gespeichert.

3.1.3. Exportfunktion LTI-Studie

In Abbildung 3.3 ist die Einbindung der Export-Funktionalität in das Web-Interface der LTI-Studie als Screenshot dargestellt. Hier konnten die unterschiedlichen, im LTI-

Studiensystem implementierten Formulare ausgewählt und entweder als ODM-XML-Datei, oder als CSV-Datei exportiert werden.

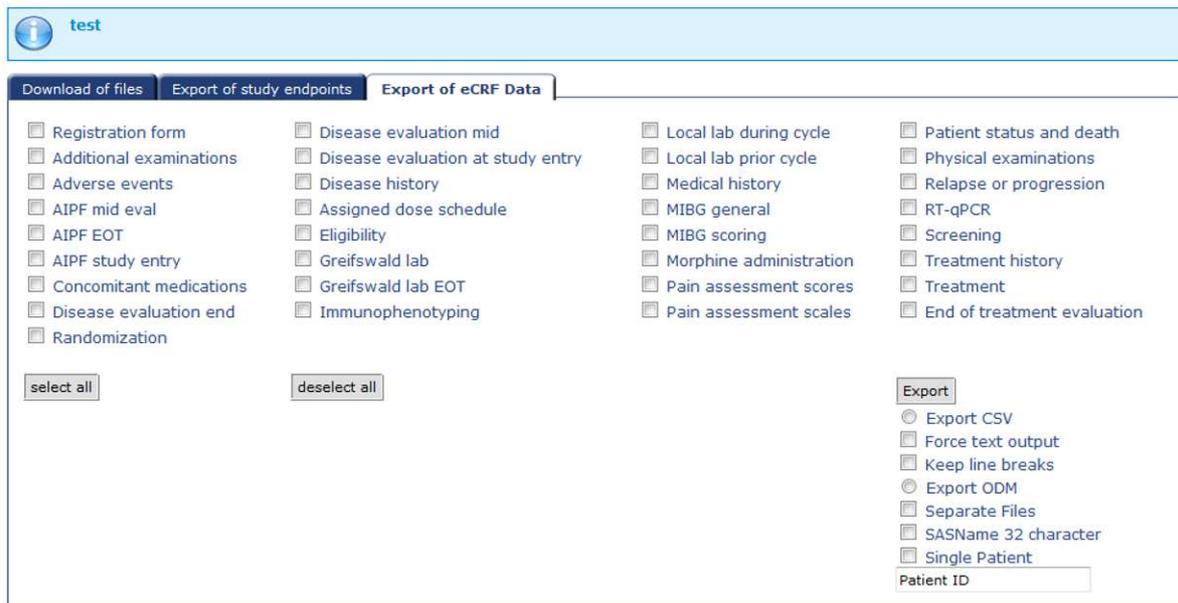


Abbildung 3.3.: Exportfunktion der LTI-Studie

Für den ODM-Export konnten zusätzlich unterschiedliche Optionen ausgewählt werden:

- Separate Files
- SASName 32 character
- Single Patient

Wurde die Option *Separate Files* ausgewählt, so wurde für jedes selektierte Formular ein XML-Dokument angelegt.

Wurde *SASName 32 character* ausgewählt, so wurden die in Kapitel 2.3.1 beschriebenen Attribute *SASDatasetName* und *SASFieldName* mit einer maximalen Länge von 32 Zeichen erstellt. Der Export von einzelnen Patienten konnte unter der Angabe des Pseudonyms des Patienten und der Auswahl der Option *Single Patient* durchgeführt werden. Für den Fall, dass keine Option ausgewählt wurde, wurde eine XML-Datei erzeugt, die alle Tabellen und Patienten im ODM-Format enthielt.

3.1.4. Ablauf

In Abbildung 3.4 ist der Ablauf des ODM-Exports grafisch dargestellt. In der Exportfunktion des LTI-Studiensystems wurde zuerst das Format des Exports gewählt. Für den ODM-Export öffnete sich ein Fenster, in dem ein Thread gestartet wurde. Dazu wurde das *threading-Modul* verwendet [44].

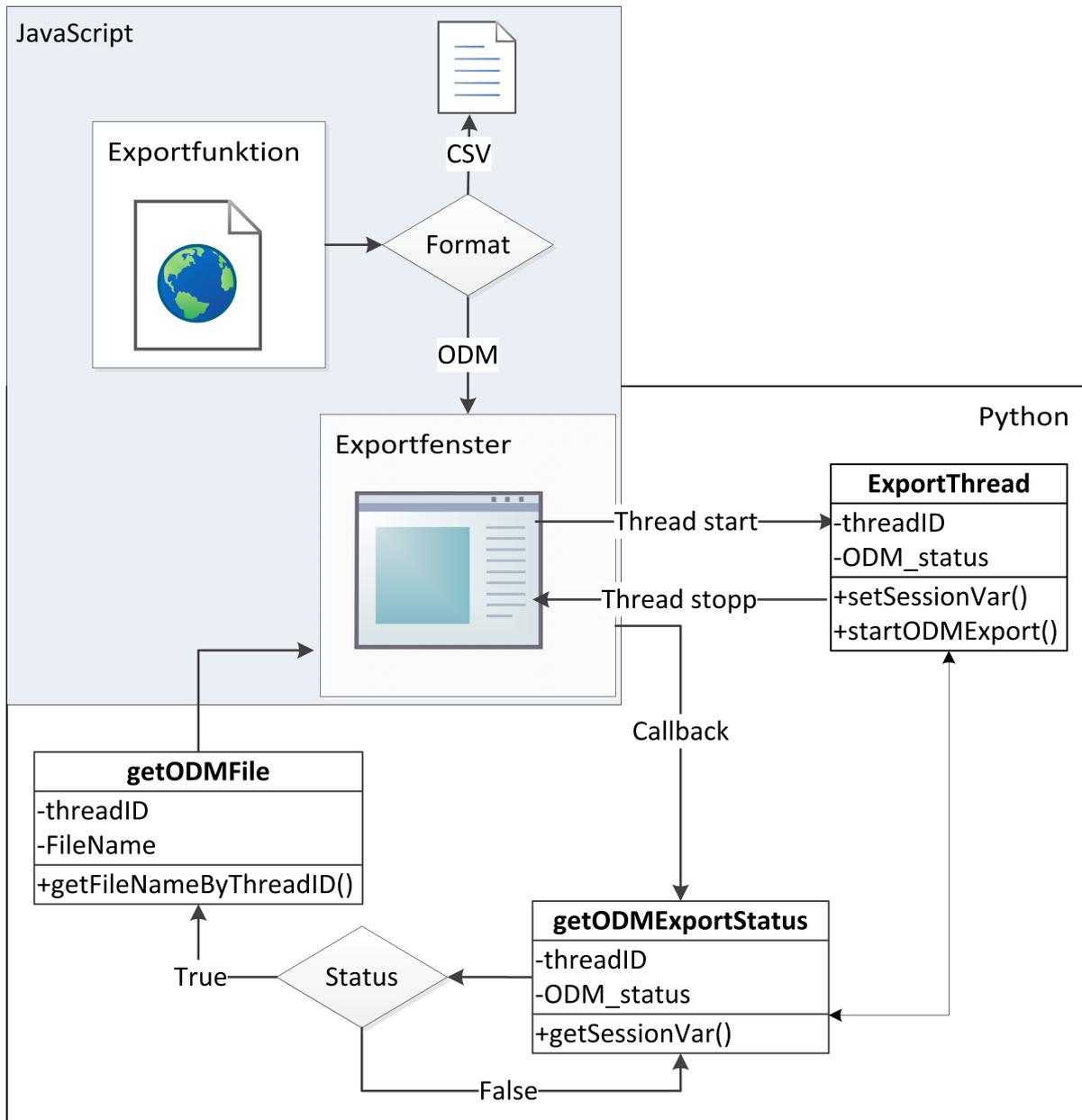


Abbildung 3.4.: ODM-Export

Mit Start des Threads wurde eine Session-Variable gesetzt, die den Status des ODM-Exports speicherte. Zu Beginn wurde diese auf *False* gesetzt.

Im Exportfenster wurde mit der JavaScript-Methode *SetInterval* regelmäßig ein Callback aufgerufen, der den aktuellen Status des ODM-Exports überprüfte. So lange dieser *False* zurückgab, wurde die Schleife weiter ausgeführt und ein Spinner angezeigt.

Konnte der Export erfolgreich durchgeführt werden, so wurde die Session-Variable *ODM_status* schlussendlich auf *True* gesetzt und, basierend auf der ID des Threads, die gezippten ODM-XML-Datei(en) abgeholt und über einen Link zum Download bereitgestellt.

3.1.5. Python Skript

Der Aufbau des Python Skripts wird in Abbildung 3.5 grafisch dargestellt. Aus dem Studiensystem heraus wurde eine studienspezifische Klasse *LTIODMExport* instanziiert und folgende Parameter angelegt:

- Feldname Patienten-ID (*SIOPEN-R-NET patient ID*)
Mit Hilfe dieses Parameters konnte das Pseudonym des Patienten im Attribut *SubjectKey* gespeichert werden.
- Identifikator für wiederkehrende *StudyEvents* (*Cycle*)
Basierend auf diesem Parameter konnten wiederkehrenden *StudyEvents* identifiziert und die *StudyEventRepeatKeys* gesetzt werden.

Durch diese Parameter konnte in den Klassen, welche die ODM-Struktur erzeugten, eine Unterscheidung der *StudyEvents* und eine Identifizierung der Patienten-ID vorgenommen werden.

Die Klasse *LTIODMExport* erbt alle Methoden von der *ODMBase*. Somit wurde vom Studiensystem aus die Methode *runODMExport* getriggert.

In der Klasse *ODM* wurden je nach gewählter Option eine oder mehrere XML-Dateien und das Grundgerüst der ODM-Baumstruktur erzeugt. Für die Erstellung der Hierarchie wurde das Paket *lxml* verwendet [45].

Als nächsten Schritt wurde in der Klasse *Subject* für jeden Patienten ein Element *SubjectData* erzeugt und, basierend auf das in der Klasse *LTIODMExport* definierte Feld für die Patienten-ID, das Pseudonym des Patienten als Attribut *SubjectKey* hinzugefügt.

In weiterer Folge wurde für jedes ausgewählte Formular ein *StudyEvent* erstellt. In dieser Klasse wurden die Elemente *StudyEventRef*, *StudyEventDef* und *StudyEventData* parallel angelegt. Des Weiteren wurde das Vorhandensein des in *LTIODMExport* definierten Indikators für wiederkehrende *StudyEvents* überprüft. Für den Fall, dass ein *Cycle* in der Tabelle existierte, wurde für jedes Auftreten dieses *StudyEvents* ein Element *StudyEventData* angelegt und mit der internen laufenden Nummer *Cycle* als *StudyEventRepeatKey* versehen.

Die Tags für die Formulare, die einem *StudyEvent* zugeordnet sind, wurden in der Klasse *Form* erstellt. Hier wurden für jedes Formular die Elemente *FormRef*, *FormDef*, *ItemGroupRef*, *ItemGroupDef*, *FormData* und *ItemGroupData* zur Baumhierarchie hinzugefügt. Es wurde hierbei davon ausgegangen, dass jedes Formular aus nur einer *ItemGroup* besteht.

Die dazugehörigen *Items* wurden in der Klasse *Item* erstellt. In dieser Klasse wurden sowohl *ItemRef*, als auch *ItemDef* und *ItemData* hinzugefügt. Im Attribut *Value* des *ItemData* Tags wurden die Daten aus der Datenbank gespeichert.

Zum Schluss wurden die *CodeLists* für jene *Items*, für die eine begrenzte Anzahl an möglichen Werten vorhanden waren, in der Klasse *CodeList* erstellt.

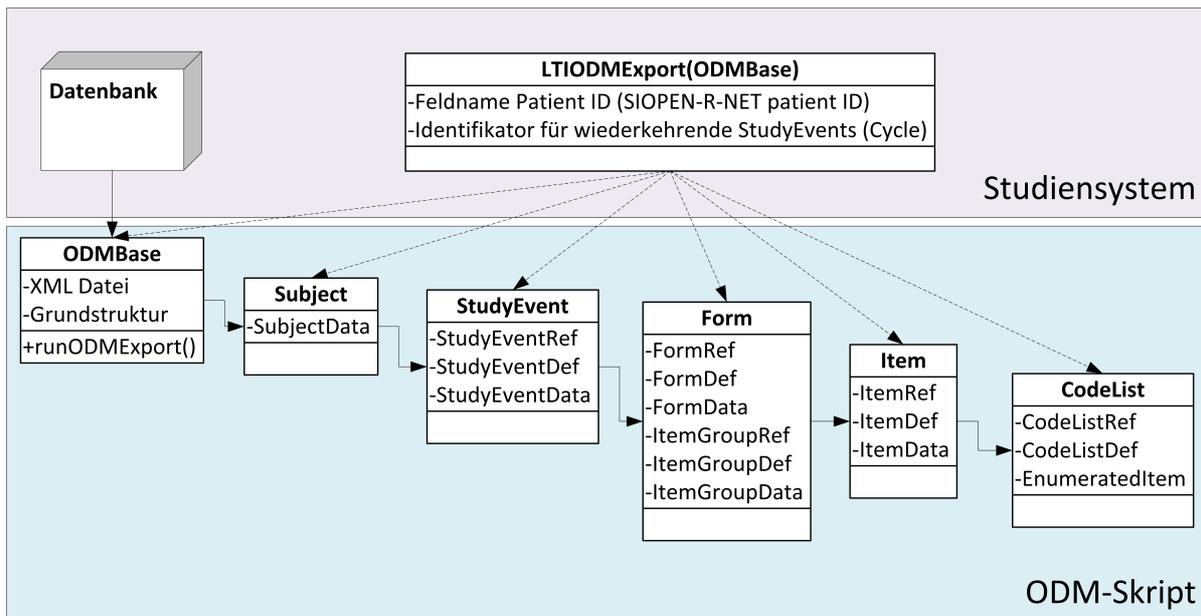


Abbildung 3.5.: Grafische Darstellung des ODM-Export-Python-Skripts. Für jedes im ODM-Format vorkommende Element wurde eine eigene Klasse implementiert, die die entsprechenden Tags und Attribute zur Baumhierarchie hinzufügte.

3.1.6. Validierungsfunktion

Um gewährleisten zu können, dass die Daten während des ODM-Exports weder vertauscht noch verfälscht wurden, wurde eine Validierungsfunktion in Python implementiert.

Diese Validierungsfunktion verglich die Patientendaten aus dem ODM-XML-Dokument mit den Daten aus der Datenbank. Dabei wurde der Parser vom Package *lxml* [45] verwendet. Basierend auf dem Pseudonym des Patienten, das sich im Attribut *SubjectKey* befand, wurde der entsprechende Eintrag des Patienten aus der Datenbank geholt. Über die *OID* des *ItemData* Elements, konnte das entsprechende Feld in der Datenbank gefunden werden. Der Wert dieses Feldes wurde dann mit dem Wert des Attributs *Value* des *ItemData* Tags verglichen.

Dieser Vorgang wurde für jeden Patienten für jedes *Item* durchgeführt und in einem Validierungsprotokoll im XML-Format festgehalten.

Im Listing 3.1 ist ein Auszug aus einem Validierungsprotokoll dargestellt. Das Kindelement von *ValidationProtocolODM* enthielt den Name des Formulars. Die ID des

Patienten wurde in das Kindelement des Formulars geschrieben. In den hier wiederum enthaltenen Subelementen sind die einzelnen *Items*, die validiert wurden, aufgelistet. Der Vergleich der Werte aus der Datenbank und aus der ODM-Datei wurde diesem Element als Text hinzugefügt. Waren die Werte der *Items* ident, so wurde das Ergebnis dem Element *Successful* hinzugefügt. Für den Fall, dass ein Fehler auftrat wurde ein Element *Error* angelegt. Dieses Element wurde im Listing 3.1 zu Demonstrationszwecken händisch hinzugefügt.

Listing 3.1: Beispiel eines Validierungsprotokolls.

```
<ValidationProtocolODM >
  <Registration >
    <IT-0493-BOV-KU>
      <Successful = "additional_personal_data_hospital">Gaslini Children's Hospital
equal to Gaslini Children's Hospital</Successful >
      <Error = "personal_data_date_of_birth">2006-05-18 not equal to 1999-04-05</Error
      >
    </IT-0493-BOV-KU>
  </Registration >
</ValidationProtocolODM >
```

3.2. ODM-Transformation

Die zweite Methode, die in der vorliegenden Arbeit entwickelt wurde, diente dazu, ODM-Dateien, wie sie beispielsweise mit Hilfe des in Kapitel 3.1 beschriebenen Export-Tools aus der LTI-Studie generiert wurden, derart zu transformieren, dass ausgewählte Daten (Kerndatensätze) dieser Exportdateien in standardisierter Form extrahiert werden konnten.

3.2.1. Kerndatensatz

In ENCCA wurden Kerndatensätze definiert, die aus den ODM-XML-Quelldokumenten extrahiert werden sollten, um semantische Interoperabilität gewährleisten zu können (siehe Kapitel 2.7). Zukünftige Studien sollen auf diese Kerndatensätze aufbauen, wodurch gewährleistet sein wird, dass aus jedem Studienexport immer eindeutig der Kerndatensatz extrahiert werden kann. Für bereits bestehende Studien war das jedoch nicht immer der Fall, weshalb ein Mapping der Studiendaten auf die Kerndatensatz-Definitionen durchgeführt werden musste. Bei der Transformation wurden dann die *Items* aus dem Quelldokument umbenannt und die *CodeListen* entsprechend der im Kerndatensatz definierten Werte angepasst.

In Tabelle 3.1 sind die bei der Transformation beteiligten Dokumente erläutert.

Tabelle 3.1.: Terminologie der für die ODM-Transformation benötigten Dateien inklusive Beschreibung und Dateiformat.

Datei	Beschreibung	Format
Kerndatensatz-Definitionen	Daten, die aus den ODM-XML-Quelldateien extrahiert wurden. Es konnte sich hierbei um einen Kerndatensatz vom Typ KDKS oder KDBP handeln.	Excel
Kerndatensatz für klinische Studien (KDKS)	Daten, die aus klinischen Studien extrahiert wurden.	Excel
Kerndatensatz für Bioproben (KDBP)	Daten, die aus Biobanken extrahiert wurden.	Excel
Kerndatensatz-Template (KS/BP)	Struktur des Kerndatensatz-Mappings.	XML
Kerndatensatz-Mapping (KS/BP)	Abbildung aller <i>SourceItems</i> zu den entsprechenden Elementen in der Kerndatensatz-Definition. Dies konnte entweder eine KDKS- oder eine KDBP-Mapping-Datei sein.	XML
ODM-XML-Quelldateien	ODM-formatierte Exportdatei des bezogenen Quellsystems.	XML (ODM)
KDKS/KDBP ODM-XML-Datei	ODM-formatiertes Ergebnis des XSLT-Transformation.	XML (ODM)

Damit der Kerndatensatz für die XSL-Transformation verwendet werden konnte, mussten die Information zunächst in eine einheitliche Form gebracht werden. Dazu wurde ein Kerndatensatz-Template entwickelt (siehe Listing 3.2). Für jeden Kerndatensatz-Typ wurde ein Template dieser Form erstellt. Im Template entsprachen die Elemente *TargetItemID* und *TargetValue* den von den ENCCA-Partnern zur Verfügung gestellten Kerndatensatz-Definitionen.

Listing 3.2: Kerndatensatz-XML-Template am Beispiel vom Kerndatensatz für klinische Studien (KDKS).

```
<Definition SourceIdentifier="Study name" SourceVersion="1.0" TargetIdentifier="
  ClinicalCoreDataset" TargetVersion="1.0">
  <Items>
    <Item SourceItemID="####" TargetItemID="Age at diagnosis" TargetFormat="integer">
    </Item>
    <Item SourceItemID="####" TargetItemID="Sex at birth" TargetFormat="integer">
      <Value SourceValue="####" TargetValue="1" TargetValueDescription="male"></
        Value>
      <Value SourceValue="####" TargetValue="2" TargetValueDescription="female"></
        Value>
    </Item>
  </Items>
</Definition>
```

Basierend auf diesem Template wurde einmalig für jede mögliche Datenquelle manuell eine Zuordnung der entsprechenden Elemente im Quelldokument von einem Zuständigen durchgeführt. In Tabelle 3.2 wird erläutert, wie die Zuordnung der Elemente durchgeführt wurde.

Tabelle 3.2.: Definition der Source Elementen des Kerndatensatz-Templates

Kerndatensatz Element	Beschreibung
SourceIdentifier	Entsprach dem ODM-Element <i>StudyOID</i> des Quelldokumentes.
SourceVersion	Definierte die Versionsnummer der Mapping-Datei.
SourceItemID	Entsprach dem ODM-Attribut <i>ItemOID</i> des ODM-Elements <i>ItemDef</i> , welches durch die <i>TargetItemID</i> ersetzt wurde.
SourceValue	Entsprach den in der <i>CodeList</i> des jeweiligen <i>Items</i> definierten erlaubten Werte. Diese wurden auf die <i>TargetValues</i> gemapped.

Im Listing 3.3 ist eine Kerndatensatz-Mapping-Datei dargestellt. Es erfolgte hier eine Zuordnung der *SourceItems* aus dem Quelldokument zu den *TargetItems* aus der Kerndatensatz-Definition. Des Weiteren wurden die Werte aus dem Quelldokument, den *TargetValues* zugeordnet. In diesem Beispiel wurden also die beiden *Items* *age* und *sex* als *Age at diagnosis* und *Sex at birth* definiert. Die Werte *Male* und *Female*, wurden wie in der Kerndatensatz-Definition beschrieben, den kodierten Werte *1* und *2* zugeordnet.

Listing 3.3: Kerndatensatz-XML-Mapping am Beispiel vom Kerndatensatz für klinische Studien (KDKS).

```
<Definition SourceIdentifier="Study name" SourceVersion="1.0" TargetIdentifier="
  ClinicalCoredataset" TargetVersion="1.0">
  <Items>
    <Item SourceItemID="age" TargetItemID="Age at diagnosis" TargetFormat="integer"></
    Item>
    <Item SourceItemID="sex" TargetItemID="Sex at birth" TargetFormat="integer">
      <Value SourceValue="Male" TargetValue="1" TargetValueDescription="male"></
      Value>
      <Value SourceValue="Female" TargetValue="2" TargetValueDescription="female"></
      Value>
    </Item>
  </Items>
</Definition>
```

3.2.2. Data Management App

Die ODM-Transformation wurde in Form einer Web-Applikation als Teil der ABCD-4-E-Plattform entwickelt. Die App wurde auf der bereits vorhandenen Struktur der ABCD-4-E weiterentwickelt (siehe Kapitel 2.6). In Abbildung 3.6 ist der Aufbau der Data Management App grafisch dargestellt. Für die Realisierung wurde Bootstrap [37] verwendet.

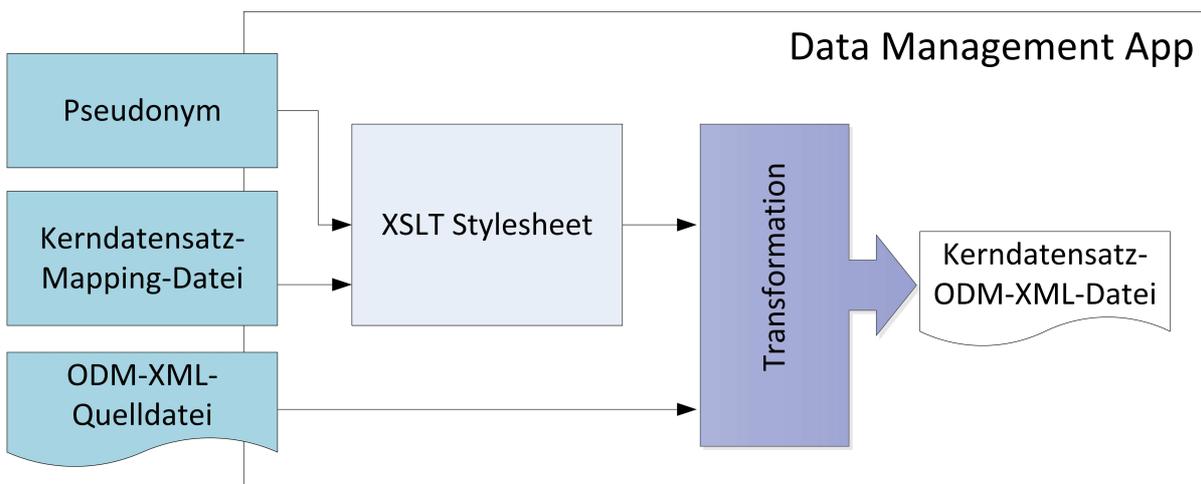


Abbildung 3.6.: Data Management App

Vom User wurden eine ODM-XML-Quelldatei und die dazugehörige Kerndatensatz-Mapping-Datei hochgeladen. Falls ein Pseudonyme im Feld *Set EUPID* angegeben wurde, so wurde diese zusammen mit dem Kerndatensatz-Mapping an den XSLT-Code als Parameter übergeben.

Als XSLT-Prozessor wurde für diese Arbeit der Saxon-CE Prozessor verwendet [46]. Dadurch konnte eine clientseitige Transformation mit Unterstützung von XSLT 2.0 und XPath 2.0 ermöglicht werden.

Im XSLT-Code wurden für jedes ODM-Element die jeweiligen Templaterregeln aufgerufen. Im ersten Schritt wurden alle Elemente, die zu den zu extrahierenden *Items* gehörten, mittels *xsl:key* Funktionen in Variablen gespeichert. So wurden für jedes *Item* die Elemente *CodeList*, *ItemDef*, *ItemGroupDef*, *FormDef*, *StudyEventDef* und *Protocol* mit dem Befehl *xsl:copy-of* kopiert. Im Tag *ItemData* wurde das Attribut *Value* durch den entsprechenden Wert aus der Kerndatensatz-Mapping-Datei ersetzt. Das vom User angegebene Pseudonym im Feld *Set EUPID* wurde verwendet, um die Patient-ID im Attribut *SubjectKey* des Quelldokuments durch die EUPID zu ersetzen (siehe Anhang A).

4. Ergebnisse

4.1. ODM-Export

Mit Hilfe des in Kapitel 3.1 beschriebenen ODM-Export-Tools wurden die Daten der LTI-Studie (Stand 2014-09-29) extrahiert und die Performance des Tools wurde analysiert. Die Auswertung erfolgte auf einen Dell OptiPlex 760 Personal Computer mit Intel(R) Core(TM)2 DUO, 4.00 GB RAM, 3.00 GHz Taktfrequenz, mit 100 MBit Internetanbindung.

In Tabelle 4.1 sind die aufgezeichneten Metadaten über den ODM-Export aufgelistet. Die Dauer eines Full Study Exports und die Größe der Full Study Export-Datei wurden mit dem Export von einzelnen Formularen verglichen. Des Weiteren wurden die Inhalte der Dateien gegenüber gestellt (siehe Tabelle 4.2).

Tabelle 4.1.: ODM-Export aus dem LTI-Studiensystem (Stand 2014-09-29). Ein Export der gesamten Studie konnte in wenigen Minuten durchgeführt werden.

Export 2015-03-20	Start	Finish	Time	File Size
Full Study	12:29:37	12:33:17	00:03:40	69,5 MB
Adverse Events	12:28:28	12:29:28	00:01:00	16,7 MB
Registration	12:33:54	12:34:04	00:00:10	100 KB

Tabelle 4.2.: Analyse der ODM-Export Dateien aus dem LTI-Studiensystem (Stand 2014-09-29). Eine Exportdatei der gesamten Studie beinhaltete rund 900.000 Zeilen.

Export 2015-03-20	# Rows	# Patients	# ItemDef	# ItemData
Full Study	889.230	124	1.429	776.330
Adverse Events	232.334	106	37	199.467
Registration	1.657	124	8	620

Ein Export der gesamten LTI-Studie konnte in einem Zeitraum von wenigen Minuten durchgeführt werden. Die fertige ODM-XML-Datei besaß eine Dateigröße von 69,5 MB. Im Vergleich konnten andere Formulare wie beispielsweise *Registration* in einem Zeitfenster von wenigen Sekunden exportiert werden. *Adverse Events* machte mit 16,7 MB den größten Teil des gesamten Exports aus. Der Export dauerte hier eine Minute.

Die Exportdatei der gesamten LTI-Studie beinhaltete 889.230 Zeilen. Basierend auf den 1.429 *Items* wurden 776.330 Patientendaten von 124 Patienten gespeichert. Davon waren 199.467 vom Formular *Adverse Events*. Nicht wiederkehrende *StudyEvents*, wie *Registration*, enthielten dementsprechend weniger *Items*, Zeilen und Daten.

Die mit Hilfe des ODM-Export-Tools exportierten ODM-Dateien wurden mit dem Validierungstool CDISC ODM Checker (siehe Kapitel 2.4) auf ihre ODM-Konformität überprüft. Es wurden keine Fehler in der ODM-Konformität identifiziert.

Des Weiteren wurde eine eigene Validierungsfunktion (siehe Kapitel 3.1.6) implementiert, die die Daten aus der ODM-XML-Datei mit den Daten aus der Datenbank verglich. Dieser Vergleich konnte erfolgreich durchgeführt werden. Dadurch wurde bestätigt, dass die Daten während des Exports weder verfälscht noch vertauscht wurden.

4.2. ODM-Transformation

Zwei Dokumente und ein optionaler Parameter waren als Eingangsgrößen bei der ODM-Transformation beteiligt:

- Die ODM-XML-Quelldatei
- Die Kerndatensatz-Mapping-Datei, die auf Basis eines Kerndatensatz-Templates für jedes ODM-XML-Quelldokument erstellt wurde
- Das EUPID-basierte Pseudonym

Listing 4.1 entspricht einem Auszug aus einer ODM-XML-Quelldatei. Die Elemente, die von der Transformation betroffen sind, wurden hervorgehoben. Die dazugehörige Kerndatensatz-Mapping-Datei ist im Listing 4.2 dargestellt.

Entsprechend dem KDKS wurden die *Items age* und *sex* in *Age at diagnosis* und *Sex at birth* umbenannt. Das *Item ethnic group* war im KDKS nicht enthalten und wurde somit auch nicht in das Ergebnisdokument kopiert.

Für das *Item sex* wurde im KDKS ein Value-Mapping definiert. Die *SourceValues Male* und *Female* wurden in die kodierten Werte *1* und *2* transformiert. Der Datentyp wurde von *text* in *integer* umgewandelt.

Falls im Textfeld *Set EUPID* ein Pseudonym angegeben wurde, so wurde die Patient-ID im Attribut *SubjectKey* durch die EUPID ersetzt. Für den Fall, dass in der ODM-XML-Quelldatei bereits eine EUPID-basierende Pseudonymisierung verwendet wurde, oder kein Pseudonym im Textfeld angegeben war, wurde die ursprüngliche Patient-ID kopiert. Im Beispiel aus Listing 4.1 und 4.3 wurde das Pseudonym *LTI 1* als nicht EUPID-konform erkannt. Für diesen Patienten wurde mit der Patient Registration App ein neues Pseudonym generiert und anschließend in der Data Management App ausgetauscht.

Das Resultat der Transformation ist im Listing 4.3 abgebildet.

Listing 4.1: Auszug aus der Input ODM-XML-Quelldatei

```

<?xml version='1.0' encoding='UTF-8'?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" ODMVersion="1.3.2
" FileOID="FD.infusion" AsOfDateTime="20014-01-18T14:01:41" FileType="Snapshot"
SourceSystemVersion="9.01.01M3D05172005" SourceSystem="SAS 9.1" CreationDateTime="
2014-10-29T07:34:19" Description="Description" xsi:schemaLocation="http://www.cdisc
.org/ns/odm/v1.3 ODM1-3-2.xsd">
<Study OID="ST.infusion">
  <GlobalVariables>
    <StudyName>Long Term Infusion</StudyName>
    <StudyDescription/>
    <ProtocolName/>
  </GlobalVariables>
  <BasicDefinitions/>
  <MetaDataVersion OID="MD.1" Name="1">
    <Protocol>
      <StudyEventRef StudyEventOID="SE.1" Mandatory="Yes"></StudyEventRef>
    </Protocol>
    <StudyEventDef OID="SE.1" Name="register patient" Type="Unscheduled"
      Repeating="No">
      <FormRef FormOID="FORM.1" Mandatory="Yes"/>
    </StudyEventDef>
    <FormDef OID="FORM.1" Repeating="No" Name="register patient">
      <ItemGroupRef ItemGroupOID="IG.1" Mandatory="Yes"/>
    </FormDef>
    <ItemGroupDef OID="IG.1" Name="personal data" Repeating="No">
      <ItemRef ItemOID="age" OrderNumber="1" Mandatory="Yes"/>
      <ItemRef ItemOID="sex" OrderNumber="2" Mandatory="Yes"/>
      <ItemRef ItemOID="ethnic group" OrderNumber="3" Mandatory="Yes"/>
    </ItemGroupDef>
    <ItemDef OID="age" DataType="integer"></ItemDef>
    <ItemDef OID="sex" DataType="text">
      <CodeListRef CodeListOID="CL.sex"/></ItemDef>
    <ItemDef OID="ethnic group" DataType="text"></ItemDef>
    <CodeList OID="CL.sex" DataType="text">
      <EnumeratedItem CodedValue="male"/>
      <EnumeratedItem CodedValue="female"/>
    </CodeList>
  </MetaDataVersion>
</Study>
<ClinicalData StudyOID="ST.1" MetaDataVersionOID="MD.1">
  <SubjectData SubjectKey="LTI 1">
    <StudyEventData StudyEventOID="SE.1">
      <FormData FormOID="FORM.1">
        <ItemGroupData ItemGroupOID="IG.1">
          <ItemData ItemOID="age" Value="3"/>
          <ItemData ItemOID="sex" Value="male"/>
          <ItemData ItemOID="ethnic group" Value="unknown"/>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>

```

Listing 4.2: Kerndatensatz-XML-Mapping vom Typ Kerndatensatz für klinische Studien (KDKS)

```

<Definition SourceIdentifier="Study name" SourceVersion="1.0" TargetIdentifier="
  ClinicalCoredataset" TargetVersion="1.0">
  <Items>
    <Item SourceItemID="age" TargetItemID="Age at diagnosis" TargetFormat="integer"></
    Item>
    <Item SourceItemID="sex" TargetItemID="Sex at birth" TargetFormat="integer">
      <Value SourceValue="Male" TargetValue="1" TargetValueDescription="male"></
      Value>
      <Value SourceValue="Female" TargetValue="2" TargetValueDescription="female"></
      Value>
    </Item>
  </Items>
</Definition>

```

Listing 4.3: Auszug aus der ODM-XML-Ergebnisatei

```

<?xml version='1.0' encoding='UTF-8'?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:xsi="http://www.w3.org/2001/
  XMLSchema-instance" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" ODMVersion="1.3.2
  " FileOID="FD.infusion" AsOfDateTime="20014-01-18T14:01:41" FileType="Snapshot"
  SourceSystemVersion="9.01.01M3D05172005" SourceSystem="SAS 9.1" CreationDateTime="
  2014-10-29T07:34:19" Description="Description" xsi:schemaLocation="http://www.cdisc
  .org/ns/odm/v1.3 ODM1-3-2.xsd">
  <Study OID="ST.infusion">
    <GlobalVariables>
      <StudyName>Long Term Infusion</StudyName>
      <StudyDescription/>
      <ProtocolName/>
    </GlobalVariables>
    <BasicDefinitions/>
    <MetaDataVersion OID="MD.1" Name="1">
      <Protocol>
        <StudyEventRef StudyEventOID="SE.1" Mandatory="Yes"/>
      </Protocol>
      <StudyEventDef OID="SE.1" Name="register patient" Type="Unscheduled" Repeating=
        "No">
        <FormRef FormOID="FORM.1" Mandatory="Yes"/>
      </StudyEventDef>
      <FormDef OID="FORM.1" Repeating="No" Name="register patient">
        <ItemGroupRef ItemGroupOID="IG.1" Mandatory="Yes"/>
      </FormDef>
      <ItemGroupDef OID="IG.1" Name="personal data" Repeating="No">
        <ItemRef ItemOID="Age at diagnosis" OrderNumber="1" Mandatory="Yes"/>
        <ItemRef ItemOID="Sex at birth" OrderNumber="2" Mandatory="Yes"/>
      </ItemGroupDef>
      <ItemDef OID="Age at diagnosis" DataType="integer"/>
      <ItemDef OID="Sex at birth" DataType="integer">
        <CodeListRef CodeListOID="CL.Sex at birth"/>
      </ItemDef>
      <CodeList OID="CL.Sex at birth" DataType="integer">
        <CodeListItem CodedValue="1">
          <Decode><TranslatedText>male</TranslatedText></Decode>
        </CodeListItem>
        <CodeListItem CodedValue="2">
          <Decode><TranslatedText>female</TranslatedText></Decode>
        </CodeListItem>
      </CodeList>
    </MetaDataVersion>
  </Study>
  <ClinicalData StudyOID="ST.1" MetaDataVersionOID="MD.1">
    <SubjectData SubjectKey="PSN-NeuroblastomaNetwork-f10f-5f6c-be9c-ed07">
      <StudyEventData StudyEventOID="SE.1">
        <FormData FormOID="FORM.1">
          <ItemGroupData ItemGroupOID="IG.1">
            <ItemData ItemOID="Age at diagnosis" Value="3"/>
            <ItemData ItemOID="Sex at birth" Value="1"/>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>

```

```

    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>

```

Die Data Management App wurde mit unterschiedlichen ODM-XML-Quelldokumenten und Mapping-Dateien getestet. Zusätzlich zur ODM-Export-Datei aus der LTI-Studie wurden von den ENCCA-IT-Partnern zwei weitere ODM-XML-Quelldokumente, mit den entsprechenden Mapping-Dateien, zur Verfügung gestellt (siehe Tabelle 4.3). Dabei handelte es sich um Dokumente aus der Neuroblastom-, Leukämie- und Wilms-Tumor-Domäne. Zwei Dokumente waren vom Typ KDKS und eines vom Typ KDBP.

Tabelle 4.3.: ODM-XML-Dateien, die von den ENCCA-IT-Partnern zur Evaluierung der Data Management App zur Verfügung gestellt wurden. (KDKS ... Kerndatensatz für klinische Studien, KDBP ... Kerndatensatz für Bioproben)

ENCCA-IT-Partner	Ursprung der ODM-Datenquelle	Typ des Kerndatensatzes	Referenz
AIT	SIOPEN-R-NET	KDKS	[13]
CINECA	Leukaemia Clinical Trials Network	KDKS	[47]
FORTH	Wilms Tumour Research Network	KDBP	[48]

In Abbildung 4.1 ist das Ergebnis der ODM-Transformation mit der ODM-XML-Quelldatei der LTI-Studie als Screenshot dargestellt. Für Demonstrationszwecke wurde eine Reihe von EUPID-konformen Pseudonymen generiert und das LTI-Export-Dokument so vorbereitet, dass für einen Patienten ein EUPID-Pseudonym angegeben werden konnte.

Für jeden Patienten wurde ein neues Dokument generiert. Während der Transformation zeigte ein Fortschrittsbalken den Fortschritt des Prozesses in Prozent an.

Nach Abschluss der Transformation wurden links die fertigen ODM-XML-Dateien ausgegeben. Rechts davon wurde der Output durch Parsen in tabellarische Form gebracht.

Transforming and Loading ...

Complete!

Result

Result Documents 10

Search

- PSN-NeuroblastomaNetwork-f10f-5f6c-be9c-ed07
- PSN-NeuroblastomaNetwork-e81e-9873-5c3f-1ee
- PSN-NeuroblastomaNetwork-2b3c-c301-5941-0e
- PSN-NeuroblastomaNetwork-e7bb-bff5-6498-0fe
- PSN-NeuroblastomaNetwork-27b5-2dc0-ec50-b5
- PSN-NeuroblastomaNetwork-1d77-6493-1185-93
- PSN-NeuroblastomaNetwork-3c3b-d0ab-d369-14
- PSN-NeuroblastomaNetwork-7ea3-dcb5-b993-03
- PSN-NeuroblastomaNetwork-9749-a8a8-2fb9-0fc
- PSN-NeuroblastomaNetwork-f10f-5f6c-be9c-ed07

PSN-NeuroblastomaNetwork-f10f-5f6c-be9c-ed07

Age in years	4
Diagnostic category	1
Date of registration	2012-08-09
Initial therapy Surgery Result	1
Date of birth	2010-06-03
Ethnic group	White or Caucasian
Sex at birth	2
Date of incidence	2011-07-01
Nmyc-amplification	1
1st Relapse/Progression Date	undefined
1st Relapse/Progression Type	9

Warnings 0

Start a new Transform & Load process

Abbildung 4.1.: Data Management App: Ergebnis der Transformation

Für *Items*, die in der Kerndatensatz-Mapping-Datei definiert wurden, jedoch nicht in der ODM-XML-Quelldatei vorhanden waren, wurde jeweils eine Meldung im Bereich *Warnings* ausgegeben.

Die ODM-Transformation konnte zusammen mit dem ODM-Export erfolgreich beim ENCCA General Assembly Meeting am 15.-16. Jänner 2015 in Brüssel demonstriert werden. Dabei wurde für einen Showcase ein virtueller Patient angelegt, für den Daten in allen von den ENCCA-Partnern zur Verfügung gestellten Dokumente aus Tabelle 4.3 vorhanden waren (Neuroblastom-Daten aus dem SIOPEN-R-NET, Leukämie-Daten aus dem Leukaemia Clinical Trials Network und Wilms-Tumor-Daten aus dem Wilms Tumor Research Network). Durch die Transformation der ODM-XML-Quelldokumente konnten diese Daten in eine einheitliche Form gebracht werden. Durch die entstandene semantische Interoperabilität können diese Daten für die sekundäre Forschung verwendet werden. Durch die Verwendung von kodierten Werten wie zum Beispiel *1* und *2* statt *Male* und *Female* soll die statistische Auswertung mit zum Beispiel Softwarelösungen von SAS [33] erleichtert werden.

Die Ergebnisse dazu wurden in einer *Proof-of-Concept-Study* veröffentlicht und für die eHealth-Konferenz 2015 in Wien eingereicht und angenommen [49].

5. Diskussion

Gerade im Bereich der pädiatrischen Onkologie, in der fast jeder Patient im Rahmen eines klinischen Studienprotokolls behandelt wird, ist der Übergang von klinischer Forschung zur Routinebehandlung nicht immer klar abgrenzbar. Der Austausch von Daten, die ursprünglich im Zuge klinischer Studien erfasst wurden, mit anderen Studiensystemen, mit Biobanken oder auch mit Krankenhausinformationssystemen wäre einerseits wesentlich für die erfolgreiche und effiziente Behandlung der Kinder, ist andererseits aber nicht unbedenklich in Hinblick auf Datenschutz und andere regulatorische Rahmenbedingungen. Das Projekt ENCCA und die darin entwickelte Plattform ABCD-4-E versuchen, Lösungen für dieses komplexe Problem zu finden. Mit der gegenständlichen Arbeit konnte ein wesentlicher Beitrag dazu geleistet werden.

Im Vorfeld wurde im Zuge von ENCCA eine Evaluierung der beiden Datenstandards ODM und CDA durchgeführt. Die Entscheidung fiel schlussendlich auf eine Anwendung des ODM, da dieses in der klinischen Forschung häufig Anwendung findet, während CDA dem gängigen Standard in der Versorgung entspricht. Des Weiteren war die weniger komplexe Struktur des ODM ausschlaggebend [50].

Durch die Implementierung des ODM-Exports können nun Daten aus der relationalen Datenbank der LTI-Studie in den für klinische Studien gebräuchlichen Datenstandard ODM exportiert werden. Dadurch wird ein Datenaustausch mit anderen klinischen Studien und Biobanken ermöglicht. Die Ergebnisse aus Tabelle 4.1 zeigen, dass ein Export von wiederkehrenden *StudyEvents* wie *Adverse Events* einen Großteil der Zeit und der Datenmenge in Anspruch nahm, während sich nicht wiederkehrende *StudyEvents* nur wenig auf die Gesamtlaufzeit des Exports auswirkten.

Das Erstellen eines vollständigen Datenexports aus dem LTI-Studiensystem dauerte 03:40 Minuten. Ein normaler HTTP-Request würde bei einer derart lang dauernden Abfrage in einen Timeout laufen. Daher wurde - wie in Kapitel 3.1.4 beschrieben - ein

Exportfenster implementiert, welches den Export in einem Thread startet und gleichzeitig kontinuierlich prüft, ob der Export fertiggestellt wurde. Dadurch konnte zwar ein Timeout verhindert werden, auch diese Lösung hat jedoch Verbesserungspotential. Wird das Exportfenster während des Exports geschlossen, so wird der Thread beendet und der User muss den Prozess somit erneut starten. Daher wäre zu überlegen, ob andere Lösungen, wie beispielsweise eine Verständigung des Users per E-mail, besser geeignet wären.

Nachdem damit zu rechnen ist, dass mit Fortbestand und Weiterführung der LTI-Studie weitere Daten in das LTI-Studiensystem eingetragen werden, dürfte auch der Export künftig noch mehr Zeit beanspruchen. Hinzu kommt, dass die LTI-Studie mit aktuell 124 Patienten im Vergleich zu anderen Studien (z.B. HRNBL-Studie, 2.639 Patienten laut Stand vom 27.3.2015) relativ klein ist. Daher kann diese Problematik künftig von noch größerer Bedeutung sein. Code-Optimierungen zur Performance-Steigerung sind daher gegebenenfalls noch durchzuführen.

Neben der Dauer für das Erstellen des Exports ist auch die Größe der Export-Datei ein limitierender Faktor für die Implementierung. Ein Export der gesamten LTI-Studie hatte eine Länge von 889.230 Zeilen, bei einer Patientenanzahl von nur 124 Patienten. ODM-XML-Dokumente bei großen klinischen Studien mit vielen Patienten können daher eine Größe im Bereich von mehreren GB erreichen. Möglichkeiten zur Kompression dieser Export-Dateien vor Download durch den User oder Übertragung an das *Repository* sollten in solchen Szenarien angedacht werden.

Es ist noch hinzuzufügen, dass beim aktuellen Export nicht der volle Umfang des ODM-Formats ausgeschöpft wurde. So könnte der Export noch um das Element *AdminData* erweitert werden. Dieses Element beinhaltet administrative Daten, wie Informationen über die im Studiensystem registrierten User. Die Notwendigkeit solch einer Erweiterung ist jedoch zum aktuellen Zeitpunkt noch nicht gegeben. Auch sind die im ODM definierten sogenannte *ReferenceData* nicht inkludiert. Diese Referenzdaten enthalten Informationen zur Interpretation der klinischen Studiendaten, wie zum Beispiel Normalbereiche. Zum einen liegt das daran, dass sich gewisse Informationen, wie Normalbereiche für *Items*, aus der Datenbank nur schwer extrahieren lassen. Zum anderen war auch hier die Notwendigkeit noch nicht gegeben.

Durch die in Kapitel 2.4 und 3.1.6 beschriebenen Validierungsmethoden konnten die ODM-Konformität der Dateien und die Fehlerfreiheit des ODM-Exports nachgewiesen werden. Sowohl das ODM Checker Tool als auch die eigens implementierte Validierungsfunktion wurden einmalig auf die Daten angewandt. Hier stellt sich die Frage, ob eine generelle Eingliederung der Validierungsmethoden in den Prozess des Exports erfolgen soll. Dazu muss das Risiko eines zukünftigen Fehlers im Export, betreffend Konformität oder Fehlerfreiheit, einer dementsprechend längeren Laufzeit des ODM-Exports gegenüber gestellt werden.

Alle Quellsysteme, deren Daten miteinander verknüpft werden sollen, müssen die Möglichkeit eines ODM-Exports besitzen. Da das für diese Arbeit entwickelte Tool stark auf das LTI-Studiensystem (Zope 2.10 und Python 2.4) zugeschnitten ist, könnte eine Einbindung des Exports in anderen Studiensysteme Anpassungen mit sich bringen.

Durch die Verwendung von Bootstrap im ganzen ABCD-4-E-Portal und die damit erreichte Responsivität, können die entwickelten Apps auch auf mobilen Geräten verwendet werden.

Die Erstellung der jeweiligen Mapping-Dateien war in einigen Fällen nicht einfach zu realisieren. Oft kann kein 1:1-Mapping, wie in dem Beispiel aus Listing 3.3, durchgeführt werden. In ENCCA ist es jedoch angedacht, dass in Zukunft jede klinische Studie in der pädiatrischen Onkologie auf die definierten Kerndatensätze aufbaut, anstatt im Nachhinein Mappings von unterschiedlich erfassten Daten durchzuführen. Durch die Verwendung der vorgegebenen *CodeListen* könnte dieses Problem in Zukunft gelöst sein.

Eine Anforderung an die Data Management App war die Möglichkeit einer vollständig clientseitigen ODM-Transformation. Aus Datenschutzgründen sollen die Daten den lokalen Client erst beim Hochladen der transformierten Dateien über eine Transaktion verlassen. Das ist notwendig um dem Sicherheitskonzept, das im Vorfeld dieser Arbeit entwickelt wurde [17], gerecht zu werden. Es zeigte sich, dass XSLT als Basis für die Transformation eine geeignete Methode ist. Durch die Verwendung des Saxon-CE Prozessors konnte die Transformation vollständig clientseitig durchgeführt werden und somit der Anforderung gerecht werden. Der Saxon-CE Prozessor läuft auf allen modernen Browsern, unter der Voraussetzung, dass Javascript am Client aktiviert ist.

Eine weitere Möglichkeit die ODM-Transformation durchzuführen, wäre die Verwendung von JavaScript-Parsern. Diese müssten im Vorfeld durch eine Recherche auf ihre Browserkompatibilität miteinander verglichen werden. Des Weiteren müsste eine geeignete Möglichkeit der Generierung von neuen XML-Dateien gefunden werden. Auch hier könnten sich Browserabhängigkeiten als Herausforderung darstellen.

5.1. Outlook und Future Work

Die Kerndatensatz-ODM-Dateien, die mit der Data Management App generiert wurden, sollen in weiterer Folge über eine Transaktion in ein *Repository* hochgeladen werden. Eine Möglichkeit wäre die Realisierung über das Integrating the Healthcare Enterprise (IHE) Profil Cross-Enterprise Document Sharing (XDS). XDS ist ein technischer Leitfaden zur Umsetzung des Datenaustauschs im Gesundheitswesen [51].

Das ABCD-4-E-Portal soll in Zukunft um weitere Apps erweitert werden. Es soll eine App entwickelt werden, mit der ODM-XML-Dokumente aus dem *Repository* über einen *QueryManager* bezogen werden können. Die Dokumente können dann für eine statistische Analyse herangezogen werden.

Überlebende von Kinderkrebserkrankungen werden im Laufe ihres Lebens immer wieder mit ihrer Krankheitsgeschichte konfrontiert. Auch im Erwachsenenalter können medizinische Details bei späteren gesundheitlichen Problemen oder für die Langzeitnachsorge relevant sein. Diese Informationen sind nicht immer im ausreichenden Maß zum richtigen Zeitpunkt und am richtigen Ort verfügbar. Ein weiteres im ENCCA-Projekt entwickeltes Tool ist daher der sogenannte *Survivorship Passport*. Es handelt sich hierbei um eine Art Pass für überlebende Kinder und Jugendliche, in dem eine Zusammenfassung der Krankheitsgeschichte der Patienten enthalten ist (z.B. Risikokategorie, Medikation, Strahlentherapie etc.). Die in der vorliegenden Arbeit entwickelten Möglichkeiten zum Export von Kerndatensätzen könnten künftig auch bei der Generierung standardisierter *Survivorship Passports* angewendet werden.

5.2. Schlussfolgerung

Zusammenfassend sollen alle von ENCCA entwickelten Tools für die ABCD-4-E-Plattform und die internationale Kooperation innerhalb des Netzwerks den Prozess der Erkenntnisgewinnung beschleunigen. Durch diese Arbeit konnte ein wichtiger Beitrag dazu geleistet werden. Durch die Implementierung des ODM-Exports können die Daten aus der LTI-Studie in standardisierter Form archiviert und für einen Austausch mit anderen Quellsystemen herangezogen werden.

Die Data Management App stellt eine wichtige Komponente bei der Verknüpfung von Daten aus unterschiedlichen Quellen dar, die in Zukunft für eine sekundäre Forschung genutzt werden sollen.

Durch neue Forschungsergebnisse auf dem Gebiet der pädiatrischen Onkologie erhofft man sich die Optimierung bestehender Therapieverfahren, eine Etablierung neuer Behandlungsmöglichkeiten und deren europaweite Standardisierung sowie eine Verbesserung der Langzeitnachsorge und damit schließlich der Lebensqualität von Kindern und Jugendlichen mit Krebserkrankung.

Abkürzungsverzeichnis

ABCD-4-E Advanced Biomedical Collaboration Domain for ENCCA

AMED Agreed Minimum Essential Data

BEACON BEACON-Neuroblastoma Trial

CDA Clinical Document Architecture

CDISC Clinical Data Interchange Standards Consortium

ENCCA European Network for Cancer Research in Children and Adolescents

EUPID ENCCA Unified Patient Identifier

HL7 Health Level 7

HRNBL High Risk Neuroblastoma Study

ICD International Statistical Classification of Diseases and Related Health Problems

KDBP Kerndatensatz für Bioproben

IHE Integrating the Healthcare Enterprise

KDKS Kerndatensatz für klinische Studien

LINES Low and Intermediate Risk Neuroblastoma European Study

LOINC Logical Observation Identifiers Names and Codes

LTI Long Term Continuous Infusion ch14.18/CHO Plus s.c. Aldesleukin (IL-2) Study

ODM Operational Data Model

OID Object Identifier

OMS/DES Opsoclonus Myoclonus Syndrome/Dancing Eye Syndrome in Children
With and Without Neuroblastoma

SIOPEN International Society of Paediatric Oncology Europe Neuroblastoma

SIOPEN-R-NET International Society of Paediatric Oncology European
Neuroblastoma Research Network

SNOMED Systematisierte Nomenklatur der Medizin

XDS Cross-Enterprise Document Sharing

XML Extensible Markup Language

XPath XML Path Language

XSLT Extensible Stylesheet Language Transformations

Literatur

- [1] Statistik Austria. *Krebserkrankungen*. URL: http://www.statistik.at/web_de/statistiken/gesundheit/krebserkrankungen/index.html (besucht am 17.03.2015).
- [2] Sudhakar A. „History of Cancer, Ancient and Modern Treatment Methods“. In: *J Cancer Sci Ther* 1, Issue 2 (Dez. 2009), S. 1–4. DOI: 10.4172/1948-5956.100000e2.
- [3] Hackl M, Hanika A, Klotz J, Leitner B, Zielonke N und Statistik Austria. *Trends der Entwicklung von Krebserkrankungen in Österreich - Eine Prognose bis 2030*. Bundesministerium für Gesundheit (BMG), Jan. 2015. ISBN: 978-3-902611-86-4.
- [4] DeSantis C u. a. „Cancer treatment and survivorship statistics, 2014“. In: *CA: A Cancer Journal for Clinicians* 64, Issue 4 (Juni 2014), 252–271. DOI: 10.3322/caac.21235.
- [5] Vassal G u. a. „Challenges for children and adolescents with cancer in Europe: The SIOP-Europe agenda“. In: *Pediatr Blood Cancer* 61(9) (Apr. 2014), 1551–1557. DOI: 10.1002/pbc.25044.
- [6] Gatta G u. a. „Childhood cancer survival in Europe 1999–2007: results of EURO CARE-5-a population-based study“. In: *The Lancet Oncology* 15 (Jan. 2014), S. 35–47. DOI: 10.1016/S1470-2045(13)70548-5.
- [7] ENCCA. *European Network for Cancer Research in Children and Adolescents*. URL: <http://www.encca.eu> (besucht am 30.10.2014).
- [8] Hayat M.A. *Pediatric Cancer, Volume 1: Neuroblastoma Diagnosis, Therapy, and Prognosis*. Springer, 2012. ISBN: 978-94-007-2417-4.
- [9] Vries M, Mirjam Houtlosser M, Wit J, Engberts D, Bresters D, Kaspers G und Leeuwen E. „Ethical issues at the interface of clinical care and research practice in pediatric oncology: a narrative review of parents’ and physicians’ experiences“. In: *BMC Medical Ethics* 12, Issue 1 (Sep. 2011), S. 18. DOI: 10.1186/1472-6939-12-18.

- [10] Gaze M u. a. „Development of an electronic database for quality assurance of radiotherapy in the International Society of Paediatric Oncology (Europe) high risk neuroblastoma study“. In: *Radiotherapy and Oncology* 97 (Dez. 2010), S. 593–595.
- [11] Rossig C u. a. „Effective Childhood Cancer Treatment: The Impact of Large Scale Clinical Trials in Germany and Austria“. In: *Pediatr Blood Cancer* 60, Issue 10 (Juni 2013), 1574–1581. DOI: 10.1002/pbc.24598.
- [12] SIOPEN. *International Society of Paediatric Oncology European Neuroblastoma*. URL: <http://www.siopen.org> (besucht am 07.01.2015).
- [13] Schreier G, Messmer J, Rauchegger G, Modre-Osprian R und Ladenstein R. „A Web-based platform for interdisciplinary biomedical research“. In: *Frontiers in Bioscience* 14: 2738-2746 (Jan. 2009). DOI: 10.2741/3410.
- [14] SIOPEN-R-NET. *International Society of Paediatric Oncology European Neuroblastoma Research Network*. URL: <https://www.siopen-r-net.org> (besucht am 07.01.2015).
- [15] Adl C, Messmer J, Ambros P, Ladenstein R und Schreier G. „Kollaborative Forschung am Beispiel der web-basierten ”BiologyScientific Study im SIOPEN-R-NET“. In: *Tagungsband eHealth2007* (Juni 2007).
- [16] Schwab M. *Encyclopedia of Cancer*. Springer Berlin Heidelberg, 2012, S. 2828. ISBN: 978-3-642-16483-5. DOI: 10.1007/978-3-642-16483-5_4476.
- [17] Nitzlnader M. „Konzept eines IHE-basierten Medizinischen Forschungsnetzwerkes in der Cloud“. Magisterarb. Austria: UMIT, 2014.
- [18] Nitzlnader N und Schreier G. „Patient identity management for secondary use of biomedical research data in a distributed computing environment“. In: *Stud Health Technol Inform* 198 (2014), S. 211–218. DOI: 10.3233/978-1-61499-397-1-211.
- [19] Canuel V, Rance B, Avillach P, Degoulet P und Burgun A. „Translational research platforms integrating clinical and omics data: a review of publicly available solutions“. In: *Brief Bioinform* 16, Issue 2 (März 2015), 280–290. DOI: 10.1093/bib/bbu006.
- [20] *caBIG*. URL: <https://cabig.nci.nih.gov/> (besucht am 10.04.2015).
- [21] Mongkolwat P, Kleper V, Talbot S und Rubin D. „The National Cancer Informatics Program (NCIP) Annotation and Image Markup (AIM) Foundation model.“ In: *J Digit Imaging* 27, Issue 6 (Juni 2014), S. 692–701. DOI: 10.1007/s10278-014-9710-3.

- [22] Verhaak RG u. a. „Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1.“ In: *Cancer Cell* 17, Issue 1 (Jan. 2010), S. 98–110. DOI: 10.1016/j.ccr.2009.12.020.
- [23] Hudson TJ u. a. „International network of cancer genome projects.“ In: *Nature* 464 (Apr. 2010), 993–998. DOI: 10.1038/nature08987.
- [24] Cerami E u. a. „The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data.“ In: *Cancer Discov* 2, Issue 5 (Mai 2012), S. 401–404. DOI: 10.1158/2159-8290.CD-12-0095.
- [25] *Georgetown Database of Cancer*. URL: <https://gdoc.georgetown.edu/gdoc/> (besucht am 13.04.2015).
- [26] Shimokawa K, Mogushi K, Shoji S, Hiraishi A, Ido K, Mizushima H und Tanaka H. „iCOD: an integrated clinical omics database based on the systems-pathology view of disease.“ In: *BMC Genomics* 11, Suppl 4 (Dez. 2010), S. 19. DOI: 10.1186/1471-2164-11-S4-S19.
- [27] Malin B. *A De-identification Strategy Used for Sharing One Data Provider’s Oncology Trials Data through the Project Data Sphere® Repository*. Juni 2013. URL: <https://projectdatasphere.org/projectdatasphere/html/resources/PDF/DEIDENTIFICATION> (besucht am 10.04.2015).
- [28] Ohno-Machado L u. a. „iDASH: integrating data for analysis, anonymization, and sharing.“ In: *J Am Med Inform Assoc* 19, Issue 2 (März 2012), S. 196–201. DOI: 10.1136/amiajnl-2011-000538.
- [29] Szalma S, Koka V, Khasanova T und Perakslis ED. „Effective knowledge management in translational medicine.“ In: *J Am Med Inform Assoc* 8, Issue 68 (Juli 2010). DOI: 10.1186/1479-5876-8-68..
- [30] Health Level Seven International. *HL7 Implementation Guide for CDA® Release 2*. URL: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7 (besucht am 01.04.2015).
- [31] ELGA GmbH. *ELGA CDA Implementierungsleitfäden HL7 Implementation Guide for CDA ® R2: Allgemeiner Implementierungsleitfaden für ELGA CDA Dokumente*. URL: <http://www.elga.gv.at/index.php?id=implementierungsleitfaden> (besucht am 01.04.2015).
- [32] CDISC. *Operational Data Model (ODM), Version 1.3.2*. URL: <http://www.cdisc.org/odm-v1-3-2> (besucht am 20.01.2015).

- [33] SAS. *SAS 9.3 CDISC Procedure User's Guide*. URL: <https://support.sas.com/> (besucht am 01.09.2014).
- [34] XML4Pharma. *The XML4Pharma CDISC ODM Checker*. URL: http://www.xml4pharma.com/CDISC_ODM_Checker/ (besucht am 26.02.2015).
- [35] St. Anna Kinderkrebsforschung. *Long Term Continuous Infusion ch14.18/CHO Plus s.c. Aldesleukin (IL-2) - Full Text View - ClinicalTrials.gov*. URL: <https://clinicaltrials.gov/ct2/show/NCT01701479> (besucht am 23.03.2015).
- [36] Zope Foundation. *Zope*. URL: <http://www.zope.org/> (besucht am 25.03.2015).
- [37] Otto M, Thornton J und Bootstrap contributors. *Bootstrap, The world's most popular mobile-first and responsive front-end framework*. URL: <http://getbootstrap.com/> (besucht am 19.03.2015).
- [38] Wright S. *Pro SharePoint 2013 App Development*. Apress, 2013. ISBN: 978-1-4302-5884-1.
- [39] Pialorsi P. *Microsoft SharePoint 2013 Developer Reference*. Microsoft Press, Mai 2013. ISBN: 978-0735670716.
- [40] § 3 Abs. 6 Bundesdatenschutzgesetz. 1990.
- [41] Louis D u. a. „The 2007 WHO Classification of Tumours of the Central Nervous System“. In: *Acta Neuropathologica* 114 (Aug. 2007), S. 97–109. DOI: 10.1007/s00401-007-0243-4.
- [42] Norlin L, Fransson MN, Eriksson M, Merino-Martinez R, Anderberg M, Kurtovic S und Litton JE. „A Minimum Data Set for Sharing Biobank Samples, Information, and Data: MIABIS“. In: *Biopreservation and Biobanking* 10 (Aug. 2012), S. 343–348.
- [43] Bongers F. *XSLT 2.0 Das umfassende Handbuch*. Galileo Computing, 2005.
- [44] Python Software Foundation. *Higher-level threading interface*. URL: <https://docs.python.org/2/library/threading.html> (besucht am 02.03.2015).
- [45] *lxml - Processing XML and HTML with Python*. URL: <http://lxml.de/> (besucht am 02.03.2015).
- [46] Saxonica. *Saxon-CE Product Information*. URL: <http://www.saxonica.com/ce/index.xml> (besucht am 26.02.2015).
- [47] Cineca. *AXMR, Advanced eXtended Multicenter Research*. URL: <http://www.cineca.it/en/content/systems-clinical-trials-amr> (besucht am 05.02.2015).
- [48] Stenzhorn H u. a. „The ObTiMA system - ontology-based managing of clinical trials“. In: *Stud Health Technol Inform* 160 (2010), S. 1090–1094. DOI: 10.1016/j.ejca.2014.10.029.

-
- [49] Hochedlinger N u. a. „Standardized Data Sharing in a Paediatric Oncology Research Network – A Proof-of-Concept Study“. In: (2015).
- [50] Seerainer C. „A standards-based document sharing architecture and prototype for medical research networks.“ Magisterarb. Graz, Austria: FH Joanneum, 2012.
- [51] IHE International Inc. *IHE IT Infrastructure 5 Technical Framework Volume 4 (ITI TF-4)*. URL: http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol4.pdf (besucht am 31.03.2015).

A. XSLT Stylesheet

Listing A.1: XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xpath-default-namespace="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:ixsl="http://saxonica.com/ns/interactiveXSLT"
  xmlns:js="http://saxonica.com/ns/globalJS"
  xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:odm="http://www.cdisc.
    org/ns/odm/v1.3">

  <xsl:strip-space elements="*" />
  <xsl:output method="xml" omit-xml-declaration="no" version="1.0" encoding="UTF-8"
    indent="yes" name="xml" />

  <!-- 1 -->

  <xsl:param name="src"></xsl:param>
  <xsl:param name="patientsrc"></xsl:param>

  <xsl:variable name="Items" xpath-default-namespace="" select="$src/Definition/Items">
    </xsl:variable>

  <xsl:variable name="Subjects" xpath-default-namespace="" select="/odm:ODM/odm:
    ClinicalData/odm:SubjectData[@SubjectKey = $patientsrc]"/>
  <xsl:variable name="CLRef" select = "/ODM/Study/$IDef/CodeListRef"/>
  <xsl:variable name="CLDef" select = "key('get_CLDef_by_CLRef', $CLRef/@CodeListOID)"/>
  <xsl:variable name="IDef" xpath-default-namespace="" select = "key('get_IDef_by_ID',
    $Items/Item/@SourceItemID)"/>
  <xsl:variable name="IGDef" select = "key('get_IGD_by_ITEMOID', $IDef/@OID)"/>
  <xsl:variable name="IGRef" select = "key('get_IGR_by_OID', $IGDef/@OID)"/>
  <xsl:variable name="FormDef" select = "key('get_FD_by_IGR', $IGDef/@OID)"/>
  <xsl:variable name="SED" select = "key('get_SED_by_FR', $FormDef/@OID)"/>
  <xsl:variable name="FormRef" select = "key('get_FR_by_OID', $FormDef/@OID)"/>
  <xsl:variable name="SER" select = "key('get_SER_by_SEOID', $SED/@OID)"/>

  <!-- search all ItemRef by ItemOID = ... -->
  <xsl:key name="get_IDef_by_ID" match="/ODM/Study/MetaDataVersion/ItemDef" use="@OID"
    />

  <!-- search all ItemGroupRef by ItemRef/ItemOID = ... -->
  <xsl:key name="get_IGD_by_ITEMOID" match="/ODM/Study/MetaDataVersion/ItemGroupDef"
    use="ItemRef/@ItemOID"/>

  <!-- search all ItemGroupRef by ItemGroupOID = ... -->
  <xsl:key name="get_IGR_by_OID" match="/ODM/Study/MetaDataVersion/FormDef/ItemGroupRef"
    use="@ItemGroupOID"/>

  <!-- search all FormDef by ItemGroupRef/ItemGroupOID = ... -->
  <xsl:key name="get_FD_by_IGR" match="/ODM/Study/MetaDataVersion/FormDef" use="
    ItemGroupRef/@ItemGroupOID"/>

  <!-- search all StudyEventDef by FormRef/FormOID = ... -->
  <xsl:key name="get_SED_by_FR" match="/ODM/Study/MetaDataVersion/StudyEventDef" use="
    FormRef/@FormOID"/>

  <!-- search all StudyEventRefs by StudyEventOID = ... -->
```

```

<xsl:key name="get_SER_by_SEOID" match="/ODM/Study/MetaDataVersion/Protocol/
StudyEventRef" use="@StudyEventOID"/>

<!-- search all FormRef by FormOID = ... -->
<xsl:key name="get_FR_by_OID" match="/ODM/Study/MetaDataVersion/StudyEventDef/FormRef
" use="@FormOID"/>

<!-- search all CLRef by ItemDefOID = ... -->
<xsl:key name="get_CLRef_by_ItemOID" match="/ODM/Study/MetaDataVersion/ItemDef/
CodeListRef" use="ancestor::ItemDef/@OID"/>

<!-- search Item by CLRef = ... -->
<xsl:key name="get_Item_by_CLRef" match="/ODM/Study/MetaDataVersion/ItemDef" use="
CodeListRef/@CodeListOID"/>

<!-- search the CLRef by CodeListOID = ... -->
<xsl:key name="get_CLRef_by_CL" match="/ODM/Study/MetaDataVersion/ItemDef/CodeListRef
" use="@CodeListOID"/>

<!-- search all CodeListDef by CLRef = ... -->
<xsl:key name="get_CLDef_by_CLRef" match="/ODM/Study/MetaDataVersion/CodeList" use="
@OID"/>

<xsl:template match="ODM">
  <xsl:for-each xpath-default-namespace="" select="$Items/Item/@SourceItemID">
    <xsl:choose>
      <xsl:when xpath-default-namespace="" test="not($IDef/@OID = current())">
        <xsl:message terminate="no">
          <xsl:text>Following CoreDataset-Item is missing in source ODM File: </
xsl:text>
          <xsl:value-of xpath-default-namespace="" select="$Items/Item[
@SourceItemID=current()]/@TargetItemID" separator=","/>
        </xsl:message>
      </xsl:when>
    </xsl:choose>
  </xsl:for-each>

  <xsl:for-each select="$Subjects">
    <xsl:variable name="filename" select="concat(@SubjectKey, '.xml')"/>
    <xsl:result-document href="{ $filename }">
      <xsl:variable name="currentSubject" select="@SubjectKey" />
      <xsl:for-each select="ancestor::ODM">
        <xsl:copy>
          <xsl:copy-of select="@*" />
          <xsl:apply-templates select="Study" />
          <xsl:apply-templates select="ClinicalData">
            <xsl:with-param name="Subject" select="$currentSubject" />
          </xsl:apply-templates>
        </xsl:copy>
      </xsl:for-each>
    </xsl:result-document>
  </xsl:for-each>
</xsl:template>

<!-- ***** -->
<!-- ***** Study ***** -->
<xsl:template match="Study">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates />
  </xsl:copy>
</xsl:template>

<xsl:template match="GlobalVariables">
  <xsl:copy-of select="." />
</xsl:template>

<xsl:template match="BasicDefinitions">
  <xsl:copy-of select="." />
</xsl:template>
<xsl:template match="MetaDataVersion">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates select="Protocol"/>
    <xsl:apply-templates select="$SED"/>
  </xsl:copy>
</xsl:template>

```

```

    <xsl:apply-templates select="$FormDef" />
    <xsl:apply-templates select="$IGDef" />
    <xsl:apply-templates select="$IDef" />
    <xsl:apply-templates select="$CLDef" />
  </xsl:copy>
</xsl:template>
<xsl:template match="Protocol">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates select="$SER" />
  </xsl:copy>
</xsl:template>
<xsl:template match="StudyEventRef">
  <xsl:copy>
    <xsl:copy-of select="@*" />
  </xsl:copy>
</xsl:template>
<xsl:template match="StudyEventDef">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates />
  </xsl:copy>
</xsl:template>
<xsl:template match="FormRef">
  <xsl:if test="current()/@FormOID = $FormRef/@FormOID">
    <xsl:copy>
      <xsl:copy-of select="@*" />
    </xsl:copy>
  </xsl:if>
</xsl:template>
<xsl:template match="FormDef">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates />
  </xsl:copy>
</xsl:template>
<xsl:template match="ItemGroupRef">
  <xsl:copy>
    <xsl:copy-of select="@*" />
  </xsl:copy>
</xsl:template>
<xsl:template match="ItemGroupDef">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates xpath-default-namespace="" select="odm:ItemRef[@ItemOID =
      $Items/Item/@SourceItemID]" />
  </xsl:copy>
</xsl:template>
<xsl:template match="ItemRef">
  <xsl:copy>
    <xsl:attribute name="ItemOID">
      <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
        current()/@ItemOID]/@TargetItemID"/>
    </xsl:attribute>
    <xsl:copy-of select="@*[name()!='ItemOID']" />
  </xsl:copy>
</xsl:template>
<xsl:template match="ItemDef">
  <xsl:copy>
    <xsl:attribute name="OID">
      <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
        current()/@OID]/@TargetItemID"/>
    </xsl:attribute>
    <xsl:attribute name="DataType">
      <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
        current()/@OID]/@TargetFormat"/>
    </xsl:attribute>
    <xsl:copy-of select="@*[name()!='OID' and name()!='DataType']" />
    <xsl:apply-templates select="CodeListRef">
      <xsl:with-param xpath-default-namespace="" name="currentItem" select="current()
        /@OID" />
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

```

```

<xsl:template match="CodeListRef">
  <xsl:param xpath-default-namespace="" name="currentItem" />
  <xsl:element name="CodeListRef">
    <xsl:attribute name="CodeListOID">
      <xsl:value-of xpath-default-namespace="" select="concat('CL.', $Items/Item[
        @SourceItemID=$currentItem]/@TargetItemID)"/>
    </xsl:attribute>
    <xsl:copy-of select="*[name()!='CodeListOID']" />
  </xsl:element>
</xsl:template>

<xsl:template match="CodeList">
  <xsl:variable name="currentCLRef" select="key('get_CLRef_by_CL', @OID)"/>
  <xsl:variable name="currentItem" select="key('get_Item_by_CLRef', $currentCLRef/
    @CodeListOID)/@OID"/>
  <!-- if element has child CodeListRef -->
  <xsl:if test="$IDef/CodeListRef/@CodeListOID = @OID">
    <xsl:element name="CodeList">
      <xsl:attribute name="OID">
        <xsl:value-of xpath-default-namespace="" select="concat('CL.', $Items/Item[
          @SourceItemID=$currentItem]/@TargetItemID)"/>
      </xsl:attribute>
      <xsl:attribute name="DataType">
        <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
          $currentItem]/@TargetFormat"/>
      </xsl:attribute>
      <xsl:copy-of select="*[name()!='OID' and name()!='DataType']" />
      <xsl:for-each xpath-default-namespace="" select="$Items/Item[@SourceItemID=
        $currentItem]/Value">
        <xsl:element name="CodeListItem">
          <xsl:attribute name="CodedValue">
            <xsl:value-of select="@TargetValue"/>
          </xsl:attribute>
          <xsl:element name="Decode">
            <xsl:element name="TranslatedText">
              <xsl:value-of select="@TargetValueDescription"/>
            </xsl:element>
          </xsl:element>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:if>
</xsl:template>

<!-- ***** -->
<!-- *****Clinical Data***** -->
<xsl:template match="ClinicalData">
  <xsl:param name="Subject" />
  <xsl:copy>
    <xsl:copy-of select="*" />
    <xsl:apply-templates select="SubjectData [@SubjectKey=$Subject]"/>
  </xsl:copy>
</xsl:template>

<xsl:template match="SubjectData">
  <xsl:copy>
    <xsl:copy-of select="*" />
    <xsl:apply-templates select="StudyEventData [@StudyEventOID=$SED/@OID]"/>
  </xsl:copy>
</xsl:template>

<xsl:template match="StudyEventData">
  <xsl:copy>
    <xsl:copy-of select="*" />
    <xsl:apply-templates select="FormData [@FormOID=$FormDef/@OID]"/>
  </xsl:copy>
</xsl:template>

<xsl:template match="FormData">
  <xsl:copy>
    <xsl:copy-of select="*" />
    <xsl:apply-templates select="ItemGroupData [@ItemGroupOID=$IGDef/@OID]"/>
  </xsl:copy>
</xsl:template>

```

```

<xsl:template match="ItemGroupData">
  <xsl:copy>
    <xsl:copy-of select="*" />
    <xsl:apply-templates xpath-default-namespace="" select="*[@ItemOID = $Items/Item/
      @SourceItemID]" />
  </xsl:copy>
</xsl:template>

<xsl:template match="ItemData">
  <xsl:variable name="correspCLRef" select="key('get_CLRef_by_ItemOID',current()/
    @ItemOID)"></xsl:variable>
  <xsl:variable name="correspCLDef" select="key('get_CLDef_by_CLRef', $correspCLRef/
    @CodeListOID)"></xsl:variable>
  <xsl:copy>
    <xsl:attribute name="ItemOID">
      <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
        current()/@ItemOID]/@TargetItemID"/>
    </xsl:attribute>

    <xsl:if xpath-default-namespace="" test="not($Items/Item[@SourceItemID=current()/
      @ItemOID]/Value)">
      <xsl:copy-of select="*[name()= 'Value']" />
    </xsl:if>
    <xsl:if xpath-default-namespace="" test="$Items/Item[@SourceItemID=current()/
      @ItemOID]/Value">
      <xsl:attribute name="Value">
        <xsl:choose>
          <!-- if value is empty -->
          <xsl:when test="current()/@Value = ''">
            <xsl:value-of xpath-default-namespace="" select="$Items/Item[
              @SourceItemID=current()/@ItemOID]/Value[@SourceValue = '']/
              @TargetValue"/>
          </xsl:when>
          <!-- if value not exists -->
          <xsl:when test="current()[not(@Value)]">
            <xsl:value-of xpath-default-namespace="" select="$Items/Item[
              @SourceItemID=current()/@ItemOID]/Value[@SourceValue = '']/
              @TargetValue"/>
          </xsl:when>
          <!-- if mapping has no value mapping -->
          <xsl:when xpath-default-namespace="" test="not($Items/Item/Value)">
            <xsl:copy-of select="*[name()= 'Value']" />
          </xsl:when>
          <!-- if value mismatch -->
          <xsl:when xpath-default-namespace="" test="not($Items/Item[@SourceItemID=
            current()/@ItemOID]/Value[@SourceValue = current()/@Value])">
            <xsl:text>VALUE MISMATCH</xsl:text>
            <xsl:message>
              <xsl:text>Mismatch: </xsl:text>
              <xsl:value-of select="current()/@Value" />
              <xsl:text> not in </xsl:text>
              <xsl:value-of xpath-default-namespace="" select="$Items/Item[
                @SourceItemID=current()/@ItemOID]/Value/@SourceValue" separator=",
                " />
            </xsl:message>
            <xsl:message select="'Value Mismatch'" terminate="no"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of xpath-default-namespace="" select="$Items/Item[
              @SourceItemID=current()/@ItemOID]/Value[@SourceValue = current()/@Value]/
              @TargetValue"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:attribute>
    </xsl:if>
    <xsl:copy-of select="*[name()!='ItemOID' and name()!='Value']" />
  </xsl:copy>
</xsl:template>

<xsl:template match="ItemDataAny | ItemDataString | ItemDataInteger | ItemDataFloat |
  ItemDataDate | ItemDataTime | ItemDataDatetime | ItemDataBoolean |
  ItemDataHexBinary | ItemDataBase64Binary | ItemDataHexFloat |
  ItemDataBase64Float | ItemDataPartialDate | ItemDataPartialTime |
  ItemDataPartialDatetime | ItemDataDurationDatetime | ItemDataIntervalDatetime |

```

```

ItemDataIncompleteDatetime | ItemDataIncompleteDate | ItemDataIncompleteTime |
ItemDataURI ">
<xsl:variable name="correspCLRef" select="key('get_CLRef_by_ItemOID', current()/
@ItemOID)"></xsl:variable>
<xsl:variable name="correspCLDef" select="key('get_CLDef_by_CLRef', $correspCLRef/
@CodeListOID)"></xsl:variable>
<xsl:element name="ItemData">
  <xsl:attribute name="ItemOID">
    <xsl:value-of xpath-default-namespace="" select="$Items/Item[@SourceItemID=
current()/@ItemOID]/@TargetItemID"/>
  </xsl:attribute>
  <!-- if mapping is no value mapping -->
  <xsl:if xpath-default-namespace="" test="not($Items/Item[@SourceItemID=current()/
@ItemOID]/Value)">
    <xsl:attribute name="Value">
      <xsl:value-of xpath-default-namespace="" select="current()/text()"/>
    </xsl:attribute>
  </xsl:if>
  <xsl:if xpath-default-namespace="" test="$Items/Item[@SourceItemID=current()/
@ItemOID]/Value">
    <xsl:attribute name="Value">
      <xsl:choose>
        <!-- if value is empty -->
        <xsl:when test="current()/text() = ''">
          <xsl:value-of xpath-default-namespace="" select="$Items/Item[
@SourceItemID=current()/@ItemOID]/Value[@SourceValue = '']/
@TargetValue"/>
        </xsl:when>
        <!-- if value not exists -->
        <xsl:when test="current()/not(text())">
          <xsl:value-of xpath-default-namespace="" select="$Items/Item[
@SourceItemID=current()/@ItemOID]/Value[@SourceValue = '']/
@TargetValue"/>
        </xsl:when>
        <!-- if mapping has no value mapping -->
        <xsl:when xpath-default-namespace="" test="not($Items/Item/Value)">
          <xsl:copy-of select="@*[name() = 'Value']" />
        </xsl:when>
        <!-- if value mismatch -->
        <xsl:when xpath-default-namespace="" test="not($Items/Item[@SourceItemID=
current()/@ItemOID]/Value[@SourceValue = current()/text()])">
          <xsl:text>VALUE MISMATCH</xsl:text>
          <xsl:message>
            <xsl:text>Mismatch: </xsl:text>
            <xsl:value-of select="current()/text()" />
            <xsl:text> not in </xsl:text>
            <xsl:value-of xpath-default-namespace="" select="$Items/Item[
@SourceItemID=current()/@ItemOID]/Value/@SourceValue"
              separator=", " />
          </xsl:message>
          <xsl:message select="'Value Mismatch'" terminate="no"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of xpath-default-namespace="" select="$Items/Item[
@SourceItemID=current()/@ItemOID]/Value[@SourceValue = current()/text()]/
@TargetValue"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
  </xsl:if>
  <xsl:copy-of select="@*[name() != 'ItemOID' and name() != 'Value']" />
</xsl:element>
</xsl:template>
</xsl:transform>

```