Konzeption, Implementierung und Qualitätsbewertung der Integration semantischer klinischer Daten aus heterogenen Registerdatenbanken in ein zentrales Datawarehouse

Inauguraldissertation
zur Erlangung des Grades eines Doktors der Humanbiologie
des Fachbereichs Medizin
der Justus-Liebig-Universität Gießen

vorgelegt von Stöhr, Mark Rainer
aus Büdingen

Gießen 2023

Aus dem Fachbereich Medizin der Justus-Liebig-Universität Gießen

Zentrum für Interstitielle und Seltene Lungenerkrankungen, Med. Klinik II

Betreuer: Prof. Dr. Günther

Gutachter: Prof. Dr. Sohrabi

Tag der Disputation: 11. Mai 2023

Inhalt

1. Einleitung

1.1. Gegenstand

Das Deutsche Zentrum für Lungenforschung (DZL) ist ein Konsortium von 29 Forschungseinrichtungen an fünf Standorten ARCN (Borstel, Lübeck, Kiel und Großhansdorf), BREATH (Hannover), CPC-M (München), TLRC (Heidelberg), UGMLC (Gießen und Marburg). Als eines von sechs Deutschen Zentren der Gesundheitsforschung (DZG) und gefördert durch das Bundesministerium für Bildung und Forschung (BMBF) werden neue innovative Therapien für Patienten mit Lungenerkrankungen entwickelt [1]. Insgesamt werden acht verschiedene Krankheitsbereiche standortübergreifend erforscht.

Um Verbundsforschung effizienter zu gestalten, wurde 2016 ein zentrales Data Warehouse am Standort UGMLC installiert. Hier fließen Daten aus allen dem DZL zugehörigen oder assoziierten Forschungseinrichtungen ein, um gemeinsam ausgewertet zu werden. Eine hierfür entwickelte Client-Software automatisiert den Datentransfer der einzelnen Quellsysteme zum zentralen Data Warehouse. Die Daten werden dabei doppelt pseudonymisiert und mit Hilfe einer auf das Quellsystem zugeschnittenen Konfigurationsdatei in ein einheitliches DZL-Schema transformiert.

Neben dem Thema Verbundsforschung im DZL liegt der Fokus dieser Arbeit des Weiteren auf dem Themenbereich „Metadaten“. Metadaten dienen der Spezifizierung und Kontextisierung von (Instanz-) Daten und können eine automatisierte Verarbeitung ebendieser ermöglichen [2]. So würde beispielsweise im Fall der Datenintegration des DZL anhand der Metadaten festgestellt, dass ein lokal erfasster Parameter dem zentral gespeicherten Parameter genau entspricht, sodass er zentral mit anderen integrierten Parametern gleicher Art zusammengeführt werden kann. Dieses zentrale Schema des DZL, welches die zentral erfassten Parameter exakt beschreibt, wird fortan als DZL Metadatenkatalog bezeichnet.

Zu Beginn dieser Arbeit existierten Ansätze, um diesen zentralen Metadatenkatalog zu verwirklichen, beispielsweise in Form einer Phänotypen- und Spezimenliste, die von der DZL-Plattform „Biobanking & Datenmanagement“ ausgearbeitet wurden. Die Informationen lagen in Form von Excel-Sheets vor und sind im Anhang dargestellt.

1.2. Bedeutung

Im Bereich der Verbundforschung erfüllt das DZL mehrere Rollen. Als integrierendes Zentrum gilt es Daten zusammenzuführen, die sich in technischer Infrastruktur und Ausführung, aber auch auf inhaltlicher Ebene stark unterscheiden. Als eines von sechs DZGs wiederum muss das DZL die Daten in einer Art und Weise halten und bereitstellen, die die Kollaboration mit anderen Zentren ermöglicht. Dies erfolgt beispielsweise durch die Einhaltung internationaler Standards für Notation und Programmier-Schnittstellen. Neben dem DZL gibt es zudem weitere internationale Lungenforschungszentren. Auch für eine Kooperation mit diesen Institutionen gilt es, die gesammelten Daten standardisiert und nachhaltig zu erfassen.

Metadaten spielen hier eine entscheidende Rolle. Sie ergänzen die rohen Daten um Informationen, die eine eindeutige Interpretation erlauben, beispielsweise durch die Angabe der Messmethode, der verwendeten Einheit, der Fixierungsart bei Biomaterialien oder auch der angewandten Instrumente bei bildgebenden Verfahren. Ausreichend definierte Metadaten ermöglichen es, Angaben verschiedener Quellen zu vergleichen und zusammenzuführen, um sie schlussendlich gemeinsam auszuwerten [2,3]. Die Verwendung internationaler Standard-Terminologien wie beispielsweise Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) oder Logical Observation Identifiers Names and Codes (LOINC) unterstützt darüber hinaus Interoperabilität zwischen (klinischen) Informationssystemen und somit auch eine automatische Zusammenführung von Daten [4].

Neben dem Ausräumen möglicher Missverständnisse bei der Interpretation von erfassten Daten stellt ein umfangreicher Metadaten-Katalog auch eine Übersicht über die im DZL als relevant erachteten Parameter dar. Dies bietet zum einen Forschern des DZL Orientierung, ermöglicht aber auch externen Wissenschaftlern Einblick in die Datensammlung des DZL.

1.3. Problematik

Das DZL besteht aus dutzenden unabhängigen Akteuren, deren Datenerfassung und -speicherung heterogen und zunächst unvereinbar sind. Die eigens entwickelte Integrations-Software vereinheitlicht all diese Quelldaten beim Transfer zum zentralen Data Warehouse hinsichtlich Struktur, aber auch hinsichtlich Semantik. Dies wiederum

setzt eine umfangreiche gemeinschaftliche Terminologie voraus, welche zu Beginn dieser Arbeit nicht existierte.

Allgemein betrachtet wird das Thema „Metadaten“ seit Jahrzehnten von Wissenschaftlern erforscht und spielt in den verschiedensten Kategorien eine entscheidende Rolle, beispielsweise in der öffentlichen Verwaltung oder beim Management von Geschäftsprozessen [5,6]. Im Bereich der Medizininformatik war das Thema zwar in der Forschung präsent, aber im klinischen und Studienbetrieb oftmals nicht ausreichend praktisch umgesetzt; So sind semantische Annotationen – beispielsweise in Form eines Data Dictionary – nur selten zu sehen. Eine Interpretation der Daten ist oft nur mit Hilfe des impliziten Wissens einzelner Beteiligter möglich [7]. Dieses Wissen wiederum ist Voraussetzung für die Verknüpfung der lokalen Parameter mit den zentral definierten DZL-Parametern (genannt Matching) sowie für das Verfassen von Transformationsregeln (genannt Mapping) [2].

Erst wenn ein DZL-weiter Metadatenkatalog definiert und die individuellen Mappings der einzelnen Datenbestände formuliert wurden, kann eine harmonisierende Zusammenführung mehrerer Datenquellen in einer zentralen Datenbank erfolgen.

1.4. Stand der Wissenschaft und Technik

Bereits vor dieser Arbeit gab es viele Lösungen, um sich der oben genannten Probleme anzunehmen. Diese reichen von Standards bis hin zu produktiven Systemen. An dieser Stelle werden die wichtigsten Entwicklungen skizziert.

1.4.1. Formale Standards zur Erfassung von Metadaten

Das Resource Description Framework (RDF) [8] ist ein Standard des World Wide Web Consortium (W3C) und stellt eine Grammatik dar, mit Hilfe derer Ressourcen beschrieben werden können. Einfach ausgedrückt besteht eine Aussage aus Subjekt, Prädikat und Objekt, wobei dem Subjekt (der Ressource) eine Eigenschaft (zum Beispiel Bezeichnung) und ein entsprechender Wert (zum Beispiel „Blutdruck“) zugeordnet werden. Auf diese Art und Weise lassen sich buchstäblich alle möglichen Aussagen formalisieren. Zudem wurden für RDF mehrere Repräsentationsformen definiert, beispielsweise XML oder Turle [9].

1.4.2. Metadaten-Leitfäden

Die FAIR-Prinzipien wurden 2016 beschrieben und etablierten sich seitdem als ein Maß für wohldefinierte Metadaten. Das Akronym steht für „Findable“, „Accessible“, „Interoperable“, und „Reusable“ [10].

1.4.3. Metadatenschemas

Wenn es im Bereich der Medizininformatik um die Beschreibung von Metadaten geht, dann werden damit zwei ISO-Standards in Verbindung gebracht. Der ISO 11179-3 (Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes) [11] wurde 1994 veröffentlicht und zuletzt 2013 überarbeitet. Eine konkrete Umsetzung zeigt das Samply.MDR [12], welches unter 1.4.6 näher beschrieben wird. Der Standard ist jedoch für eine Umsetzung eines Metadaten-Repositories (Begriffsklärung in 1.4.6) im Bereich der Medizininformatik nur bedingt geeignet. Dies hat dazu geführt, dass 2019 der ISO 21526 (Health informatics — Metadata repository requirements (MetaRep)) [13] veröffentlicht wurde.

Das Simple Knowledge Organization System (SKOS) definiert eine Reihe von Prädikaten und Anwendungsregeln innerhalb von RDF [14]. Es wurde ebenfalls von W3C entwickelt und dient der standardisierten Verfassung von Thesauri. Der zuvor genannte ISO 21526 empfiehlt die Verwendung von SKOS für die Klassifikation von Inhalten, die auf Benutzeroberflächen dargestellt werden sollen.

Der Dublin Core (DC) Standard [15] definiert eine Reihe von Attributen zur Beschreibung von Objekten im Internet wie Dokumente, Audioaufnahmen, und vieles mehr. Er beschreibt diese Objekte inhaltlich, aber auch die zugehörigen Personen und Rechte, sowie die Vernetzung zu anderen Ressourcen.

Das Operational Data Model (ODM) vom Clinical Data Interchange Standards Consortium (CDISC) [16] wird an dieser Stelle repräsentativ für Standards genannt, die sich mit der Architektur von Studien, Formularen und konkreten Datenfeldern befassen.

1.4.4. Terminologien / Metadaten-Kataloge / Klassifikationen

Während die zuvor vorgestellten Metadatenschemas vorgeben wie eine Ressource beschrieben werden soll, handelt es sich bei den folgenden Standards um konkrete Ressourcen mitsamt ihren Annotationen und zugehörigem Kontext. Die Terminologien SNOMED-CT, LOINC, und International Classification of Diseases 10th Revision (ICD-

10) der Weltgesundheitsorganisation (WHO) [17] sind kontrollierte Vokabulare, welche Erkrankungen, Prozeduren, Messwerte und weitere Parameter des klinischen Umfelds definieren. Sie werden international genutzt und ihre Verwendung erlaubt die uneingeschränkte Vergleichbarkeit von Daten. Das Unified Medical Language System (UMLS) [18] ist eine Zusammenführung mehrerer der zuvor genannten kontrollierten Vokabulare, hat dabei jedoch weniger Ansprüche hinsichtlich der Datenqualität. So findet beispielsweise keine Harmonisierung der Parameter statt und diese können bei gleicher Bedeutung mehrmals definiert worden sein.

1.4.5. Metadaten-Bearbeitungssoftware

Protégé ist das bekannteste Tool, um hierarchische Metadaten im Bereich der Bio- und Medizininformatik zu erstellen und zu verwalten. Es bietet eine Benutzeroberfläche, die die wesentlichen Werkzeuge zur Bearbeitung von Thesauri bereitstellen soll [19]. Das als Basis dienende Metadaten-Schema beschränkte sich auf die Beschreibung von Klassen und Sub-Klassen.

1.4.6. Metadaten-Repositories

Ein Metadaten-Repository ist ein Register, welches Metadaten zentral vorhält und bereitstellt. Spezielle Merkmale eines Metadaten-Repositories sind das zugrundeliegende Datenformat (beispielsweise relationale Datenbanken oder Textdateien), das verwendete Schema (beispielsweise ISO 11179-3), verwendete Terminologien, angewandte Leitfäden und nicht zuletzt der entsprechende Verwendungszweck.

BioPortal [20] ist das dem zuvor beschriebenen Protégé zugehörige Metadaten-Repository. Es beinhaltet dutzende Ontologien, darunter auch Repräsentationen der zuvor genannten Terminologien SNOMED-CT, LOINC und ICD-10.

Ein Beispiel für ein deutsches Metadaten-Repository ist das Samply.MDR. Es basiert auf dem zuvor genannten ISO 11179-3 Standard und wurde im Laufe dieser Arbeit als Metadatenverzeichnis für beispielsweise das Deutsche Konsortium für Translationale Krebsforschung (DKTK) [21] und die German Biobank Node (GBN) [22] genutzt.

1.4.7. Data Warehouse Lösungen

Zuletzt werden beispielhaft Data Warehouse Lösungen beschrieben, welche der Speicherung und in eingeschränkter Form auch der Auswertung von Patientendaten dienen. Informatics for Integrating Biology and the Bedside (i2b2) [23] ist ein Open

Source Projekt aus den USA, welches in immer mehr Forschungsszenarien zum Einsatz kommt. Die Stärken von i2b2 liegen in der vielseitig einsetzbaren Datenstruktur, der intuitiven Benutzeroberfläche für einfache Datenbestandsabfragen sowie der großen internationalen Community.

Eine Alternative stellt das "Common Data Model" der Organisation "Observational Medical Outcomes Partnership" (OMOP-CDM) [24] dar. Ziel bei dessen Entwicklung war und ist ein Datenmodell zu schaffen, das systematische Analysen im Bereich der Medizin in großem Umfang unterstützt. Allerdings gibt es für dieses Datenmodell bis dato keine Benutzeroberfläche, die weniger technisch versierten Forschern dessen Nutzung ermöglicht [25].

Die Firma KAIROS GmbH bietet mit CentraXX ein Werkzeug für Biobanking und Studienmanagement. Dieser Fokus und die damit einhergehende Datenstruktur ermöglichen den effektiven Einsatz bei vielen Biomaterialbanken [26], machen das Produkt aber weniger attraktiv, wenn es um die Zusammenführung von phänotypisierenden klinischen Daten mit Informationen über Biomaterial und Bildmaterial geht. Es handelt sich hier zudem um ein kommerzielles Produkt, was dessen Einsatz für das zentrale DZL Data Warehouse weniger attraktiv macht.

1.4.8. Architektur und Informationssicherheit

Informationssicherheit im Informationsverbund Metadaten-Management

Dem Thema Informationssicherheit kommt – wie bei jeder zu implementierenden IT-Lösung – auch bei der Bereitstellung von Diensten zum Bearbeiten und Abrufen eines Metadatenkatalogs eine besondere Bedeutung zu. Da das in dieser Arbeit vorgestellte Konzept des Collaborative Metadata Repository (CoMetaR) gewisse Spielräume in der Art und Weise der Implementierung zulässt, werden an dieser Stelle diejenigen Aspekte benannt, welche hier eine besondere Rolle spielen. Im anschließenden Abschnitt wird die konkrete Realisierung im DZL näher erläutert und kann als Empfehlung verstanden werden. Folgend wird das Vokabular des Bundesamts für Sicherheit in der Informationstechnik (BSI) verwendet [27]. Zur Definition des betrachteten Informationsverbundes, welcher fortan als „Metadaten-Management" bezeichnet wird, werden zunächst die zugrundeliegenden Geschäftsprozesse, die zu schützenden Informationen sowie die eingesetzten technischen Komponenten benannt. Aus diesen leiten sich Bedrohungen ab, welche im IT Grundschutz Kompendium des BSI näher

beschrieben sind [27]. Zum Schluss werden diejenigen Bedrohungen benannt, welche speziell in dem dieser Arbeit zugrundeliegenden Szenario von Bedeutung sind.

Die dem Metadaten-Management zugrundeliegenden Geschäftsprozesse sind: (1) Die Bearbeitung und das Hochladen von Metadaten, (2) der Abruf ebenjener Metadaten über einen Webbrowser, (3) die Prüfung von definiterten Mappings in Konfigurationsdateien, welche für den Upload von patientenbezogene Daten ins Data Warehouse formuliert wurden, (4) Datenbestandsabfragen im Data Warehouse, (5) Benutzerdatenverwaltung des Metadaten-Repositories, (6) Benutzerdatenverwaltung der Metadaten-Webapplikation und (7) Softwareupdates des Metadaten-Repositories und der Web-Applikation.

Die im Informationsverbund Metadaten-Management zu schützenden Informationen sind: (1) Der Metadatenkatalog und dessen Inhalte, (2) die erfassten Informationen über Personen, die einen Beitrag zum Metadatenkatalog leisten (Benutzername, Zeitstempel der Änderung, die Änderungen selbst, die den Änderungen angehängten Kommentare), (3) die Benutzerdaten für das Metadaten-Repository, (4) die Benutzerdaten für die Metadaten-Webapplikation, (5) Informationen, die aus dem Metadatenkatalog und seinen Änderungen abgeleitet werden, beispielsweise symmetrische Relationen (broader/narrower), (6) die Konfigurationen, welche für Uploads von patientenbezogenen Daten ins Data Warehouse verwendet werden und (7) die Ergebnisse bei Abfragen an das Data Warehouse, welches die Datenstruktur vom Metadatenkatalog übernimmt.

Beim Ausführen der oben genannten Geschäftsprozesse sind mehrere technische Komponenten involviert. Um diese mit Hilfe des IT Grundschutz Kompendiums (Version 2021) zu modellieren, wird, wenn vorhanden, die entsprechende Bausteinkennung angeführt. Hierzu gehört der CoMetaR Server, welcher als Allgemeiner Server (SYS.1.1) modelliert werden kann. Eine auf diesem Server laufende Applikation ist die Repository Komponente, welche als Fileserver (APP.3.3) modelliert werden kann. Zu dieser Komponente werden via der Versionierungssoftware Git Daten von Mitarbeitern hochgeladen. Die entsprechenden Bausteine zur Modellierung des Vorgangs wären Personal (ORP.2), an die jeweilige Infrastruktur anzupassende Netzwerk-Bausteine (NET), sowie Benutzerendgeräte, welche als allgemeiner Client (SYS.2.1) modelliert werden können. Die zweite Komponente auf dem CoMetaR Server ist die Webapplikation, welche als Webserver (APP.3.2) modelliert wird. Das entsprechende

Gegenstück ist auf Benutzerseite der Webbrowser (APP.1.2) auf einem Benutzerendgerät bzw. allgemeinen Client (SYS.2.1). Auch hier sind entsprechende Netzwerk-Bausteine zu beachten, die Web-Applikation verwendet lediglich das HTTP bzw. HTTPS-Protokoll. Der Data Warehouse Server kann ebenfalls als allgemeiner Server (SYS.1.1) modelliert werden. Die Datenbankkomponente entspricht einer relationalen Datenbank (APP.4.3). Sie enthält sowohl patientenbezogene Daten als auch terminologische Metadaten. Diese Metadaten werden neu geladen, sobald Änderungen am Metadatenkatalog erfolgten. Auch die Netzwerkkommunikation zwischen den beiden Servern muss gesichert werden. Die web-basierte Abfragekomponente des Data Warehouse kann als Webserver (APP.3.2) modelliert werden. Auch hier gilt es Netzwerk (NET) und Web-Browser (APP.1.2) auf dem Benutzerendgerät bzw. allgemeinen Client (SYS.2.1) zu modellieren.

Der Metadatenkatalog durchläuft in mehreren Schritten die oben genannten IT Systeme, wobei jederzeit die Gefahr des Verlustes von Daten im Sinne einer Löschung besteht. Allerdings liegen auf den Endgeräten aller Mitarbeiter, die den Katalog bearbeiten, lokale Kopien vor. Die Gewichtung der Verfügbarkeit von Diensten obliegt dem implementierenden Unternehmen. Gegebenenfalls muss der Metadatenkatalog auch vor Verlust an andere Personen/Unternehmen geschützt werden, beispielsweise wenn es gilt geistiges Eigentum zu schützen. Das DZL wiederum ist eine Verbreitung der Informationen zwecks Interoperabilität und Wissensaustausch sogar explizit gewünscht. Eine Korrumpierung des Metadatenkatalogs hingegen kann in jedem Fall zu Schaden führen. Zwar ist der Katalog durch mehrere Prüfschritte in CoMetaR vor syntaktischen Fehlern, welche eine weitere Verarbeitung verhindern würden, geschützt, falsche Inhalte werden jedoch nicht identifiziert. Eine inhaltliche Verfälschung kann zu Missinterpretationen klinischer Parameter aufgrund falscher Annotationen über falsche/missverständliche/unvollständige Ergebnisse im Abfragetool des Data Warehouse bis hin zu Ansehensverlust der Organisation führen.

Der Verlust einer Quellsystem-spezifischen Upload-Konfiguration kann nur auf dem System des Benutzers geschehen, da daraus keine Daten an den CoMetaR Server übertragen werden. Hier wird lediglich lokal JavaScript ausgeführt.

Um Änderungen am Metadatenkatalog nachvollziehen zu können, werden gewisse Informationen zu den Änderungen gespeichert und in der CoMetaR Web Applikation zur Verfügung gestellt, beispielsweise Benutzername und Zeitstempel. Diese Informationen

sind für die Nachverfolgung von Änderungen essenziell. Mitarbeiter müssen dementsprechend vorher darüber unterrichtet werden, welche Informationen öffentlich zugänglich sind. Eine Identitätsübernahme, beispielsweise durch die Erlangung der Zugangsdaten einer Person, könnte zu gefälschten Änderungen der Metadaten führen und die Würde der entsprechenden Person angreifen.

Eine schlussendliche Bewertung des Sicherheitsrisikos für einzelne Assets obliegt dem implementierenden Unternehmen. Dasselbe gilt für die Wahl zwischen Basis-, Standard- und Kernabsicherung. Als Kritische Infrastruktur (KRITIS) ist das Metadaten-Management in keinem Fall zu betrachten, da keine Auswirkungen auf das staatliche Gemeinwesen zu befürchten sind. Im folgenden Abschnitt wird die Realisierung des Informationsverbunden „Metadaten-Management“ innerhalb des DZL näher erläutert sowie in Bezug zur Datenintegration im Allgemeinen hergestellt.

Informationssicherheit in der DZL Datenintegrations-Architektur

Im Folgenden wird die Datenintegrations-Architektur des DZL schematisch präsentiert und die Rolle des Metadaten-Repository erläutert (siehe Abbildung 1).

Kernkomponente der Datenintegration und -nutzung im DZL ist ein Data Warehouse Server (3), auf welchem die Software i2b2 eingesetzt wird. Die Datenhaltung ist zweigeteilt in ein Schema für terminologische Metadaten (zum Beispiel die Klassifikation des Parameters Forced Expiratory Volume in 1 second (FEV1)) und eines für patientenbezogene Instanzdaten (zum Beispiel einem konkreten FEV1-Wert für einen konkreten Patienten). Eine Webanwendung (3.1) erlaubt mittels graphischer Oberfläche Abfragen auf Basis der Metadaten zu formulieren (zum Beispiel „Wie viele Patienten haben Asthma und einen FEV1-Wert kleiner 30% der Forced Vital Capacity (FVC)?“), welche auf den Instanzdaten ausgeführt und deren Ergebnisse in aggregierter Form zurückgemeldet werden (zum Beispiel „327±3 Patienten erfüllen die Suchkriterien.“). Nutzerzugänge sind in einem zusätzlichen Datenbankschema realisiert. Da nur befugte Personen Zugriff zur Weboberflächen haben und diese nur aggregierte Ergebnisse zur Verfügung stellt, sind die Instanzdaten doppelt gesichert.

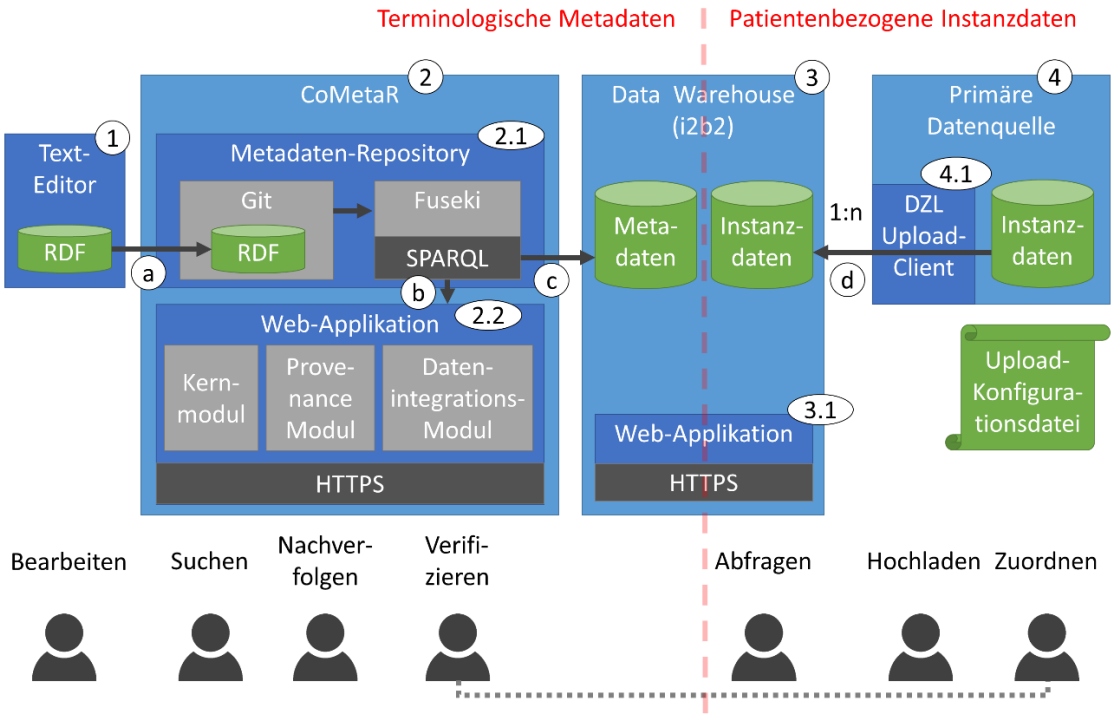


Abbildung 1: Schematische Darstellung der Datenintegrations-Architektur des DZL. Hellblauer Kasten: Server; Dunkelblauer Kasten: Anwendung; Hellgrauer Kasten: Anwendungskomponente; Dunkelgrauer Kasten: Anwendungsschnittstelle; Grüner Zylinder: Persistente Datensammlung; Untere Reihe: Anwendungsprozesse; Pfeile: Datenübertragungswege zwischen Servern, Anwendungen und Komponenten; Kreise: Technischen Komponenten (Zahlen) und Datenübertragungswege (Buchstaben), auf die in diesem Abschnitt Bezug genommen wird.

Die primären Datenquellen (4) übertragen (d) ihre patientenbezogenen Informationen mittels eines bereits vor dem Beginn dieser Arbeit eigens entwickelten Datenintegrationssoftware (4.1) an das zentrale Data Warehouse (3). Das genaue Verfahren ist im entsprechenden Datenschutzkonzept beschrieben und wurde sowohl von der Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V. (TMF) als auch von den Datenschützern der Universität Gießen und des Landes Hessen mit einem positiven Votum versehen. Bereits auf dem Rechner, auf welchem die Datenintegrationssoftware ausgeführt wird, werden die patientenidentifizierenden Informationen pseudonymisiert und verlassen den Standort nie in Rohform. Ein Pseudonymisierungsdienst eines Treuhänders in Hannover pseudonymisiert die Daten ein weiteres Mal und garantiert so, dass niemand vom zentralen Datenmanagement des DZL eine Reidentifizierung vornehmen kann. Die Daten werden verschlüsselt und doppelt

psyeudonymisiert an das zentrale Data Warehouse versendet. Sowohl zur Authentifizierung und Autorisierung als auch zur Transportverschlüsselung werden TLS-Zertifikate eingesetzt. Da TLS-Zertifikate auf Protokollebene arbeiten, wird die Rechtmäßigkeit des Datentransfers bereits beim Verbindungsaufbau sichergestellt. Voraussetzung für eine Übermittlung an das zentrale Data Warehouse ist eine gültige Patienteneinwilligung, beispielsweise der Broad Informed Consent (BIC) des DZL oder eine äquivalente Einwilligung, welche den Datenaustausch innerhalb des Forschungsnetzwerkes abdeckt. Die Prüfung auf Vorhandensein einer solchen Einverständniserklärung und eine entsprechende Filterung von Patienten vor Datenübertragung obliegt dem einzelnen Standort. An zentraler Stelle werden bei neuer Datenübermittlung alle bisherigen Daten, welche vom gleichen Zertifikat bereits früher an das Data Warehouse übermittelt wurden, gelöscht und die neuen Daten in der Datenbank gespeichert. Folglich ist eine (gezielte) Löschung von Daten durch den Standort selbst jederzeit möglich und im Falle einer zurückgezogenen Patienteneinwilligung notwendig.

Die Metadaten wiederum werden in einem Text-Editor (1) bearbeitet und an das Metadaten-Repository (2.1) gesendet. Die Versionierung der Metadaten erfolgt durch das Werkzeug Git [28]. Hierfür wird ein HTTP-Backend verwendet, die Daten werden von einem Webserver per HTTPS Port 443 entgegengenommen und an die Git-Anwendung weitergeleitet. Die Benutzerverwaltung, also Authentifizierung und Autorisierung von Nutzern, erfolgt ebenfalls in der Webserver-Komponente. Hierfür wird ein Nginx Server verwendet. Für die weitere Verwendung der Metadaten, beispielsweise die Anzeige in unserer Webapplikation (2.2,b) oder die Übernahme in i2b2 (3,c), müssen Abfragen mit Hilfe der SPARQL Protocol And RDF Query Language (SPARQL) [29] Schnittstelle an den Apache Jena Fuseki Server erfolgen. Auch hier dient der Webserver als Proxy, wodurch das gesamte Metadaten-Repository nach außen lediglich Port 443 öffnen muss.

Sowohl der CoMetaR- als auch der i2b2-Server werden vom Hochschulrechenzentrum in Gießen gehostet, wodurch nötige Sicherheitsstandards sowie eine regelmäßige Datensicherung gewährleistet sind.

2. Zielsetzung

Ziel dieser Arbeit ist es, den Experten unterschiedlicher Fachbereiche des DZL zu ermöglichen, einen Metadaten-Katalog für den Bereich der Lungenforschung zu entwickeln. Klinisches Personal und Forscher, aber auch Dokumentare und Datenmanager müssen den Metadaten-Katalog durchsuchen und für sie relevante Informationen finden können. Der Katalog muss interoperabel gestaltet sein, um einen Austausch mit anderen Software-System zu erlauben, beispielsweise anderen DZGs oder anderen Lungenforschungszentren. Die Metadaten müssen wiederverwertbar sein, insbesondere für den Bereich der Lungenforschung. Dies beinhaltet auch das Bereitstellen der Änderungs-Historie (Stichwort Provenance). Allen Akteuren des DZL soll prinzipiell das Mitwirken am Katalog ermöglicht werden. Dies beinhaltet sowohl das Erstellen und Editieren als auch das Rückmelden von Unstimmigkeiten.

2.1. Fragestellungen

Aus den oben genannten Voraussetzungen und Zielen ergeben sich die folgenden Fragestellungen, welche in den angefügten Manuskripten beantwortet wurden:

1. Es existieren verschiedene Entwicklungen und Standards zur (a) Notation, (b) Speicherung und (c) Kommunikation von Metadaten. Welche dieser Entwicklungen begünstigen die unter Zielsetzung genannten Voraussetzungen?
2. Wie kann erreicht werden, dass der erarbeitete Metadaten-Katalog bestmöglich hinsichtlich (Baum-) Struktur und Informationsumfang im Data Warehouse (i2b2) und dessen Abfrage-Tool widergespiegelt wird?
3. Verschiedene Personengruppen haben Interesse an der Mitwirkung bei der Entwicklung des DZL Metadaten-Katalogs. Wie kann gewährleistet werden, dass all jene Personen entsprechende Anpassungen vornehmen können?
4. Wie kann der aktuelle Stand des Metadaten-Katalogs visuell dargestellt werden, sodass alle interessierten Personen (Forscher, Ärzte, Dokumentare, Datenmanager, …) die für sie relevanten Informationen finden?
5. Wie kann geprüft werden, ob die patientenbezogenen Daten individueller Quellsysteme dem DZL Metadatenschema bei Übermittlung an das zentrale Data Warehouse korrekt zugeordnet wurden?
6. Der DZL Metadaten-Katalog befindet sich in stetiger Entwicklung. Wie können vergangene Änderungen am Katalog visuell dargestellt werden?

3. Übersicht zu den Manuskripten

Im folgenden Abschnitt werden die publizierten Originalarbeiten aufgelistet. Insgesamt liegen dieser Arbeit sieben Originalarbeiten zugrunde. Für jede dieser Arbeiten gilt, dass Konzeption, Ausführung und Publikation im Wesentlichen durch Mark Stöhr erstellt und durchgeführt wurden. Die Koautoren unterstützten hierbei beratend durch kritische Rückmeldungen bei regelmäßigen Besprechungen.

1. M.R. Stöhr, R.W. Majeed, A. Günther, *CoMetaR: A Collaborative Metadata Repository for Biomedical Research Networks*, Studies in health technology and informatics **245** (2017), 1337.
2. M.R. Stöhr, R.W. Majeed, A. Günther, *Using RDF and Git to Realize a Collaborative Metadata Repositoy*, Studies in health technology and informatics **247** (2018), 556–560.
3. M.R. Stöhr, R.W. Majeed, A. Günther, *Metadata Import from RDF to i2b2*, Studies in health technology and informatics **253** (2018), 40-44.
4. M.R. Stöhr, A. Günther, R.W. Majeed, *Provenance for Biomedical Ontologies with RDF and Git*, Studies in health technology and informatics **267** (2019), 230-237.
5. M.R. Stöhr, A. Günther, R.W. Majeed, *Verifying Data Integration Configurations for Semantical Correctness and Completeness*, Studies in health technology and informatics **267** (2019), 66-73.
6. M.R. Stöhr, A. Günther, R.W. Majeed, *ISO 21526 Conform Metadata Editor for FAIR Unicode SKOS Thesauri*, Studies in health technology and informatics **278** (2021), 94-100.
7. M.R. Stöhr, A. Günther, R.W. Majeed, *The Collaborative Metadata Repository (CoMetaR) Web App: Quantitative and Qualitative Usability Evaluation*, JMIR medical informatics **9** (2021), e30308.

Das folgende Schaubild (Abbildung 2) stellt einen schematischen Aufbau der Metadaten-Management-Infrastruktur dar. Mit roten Kreisen wurden jene Komponenten markiert, welche von einem genannten Manuskript behandelt wurden. Die Nummerierung entspricht dabei der Aufzählung zu Beginn dieses Abschnitts.

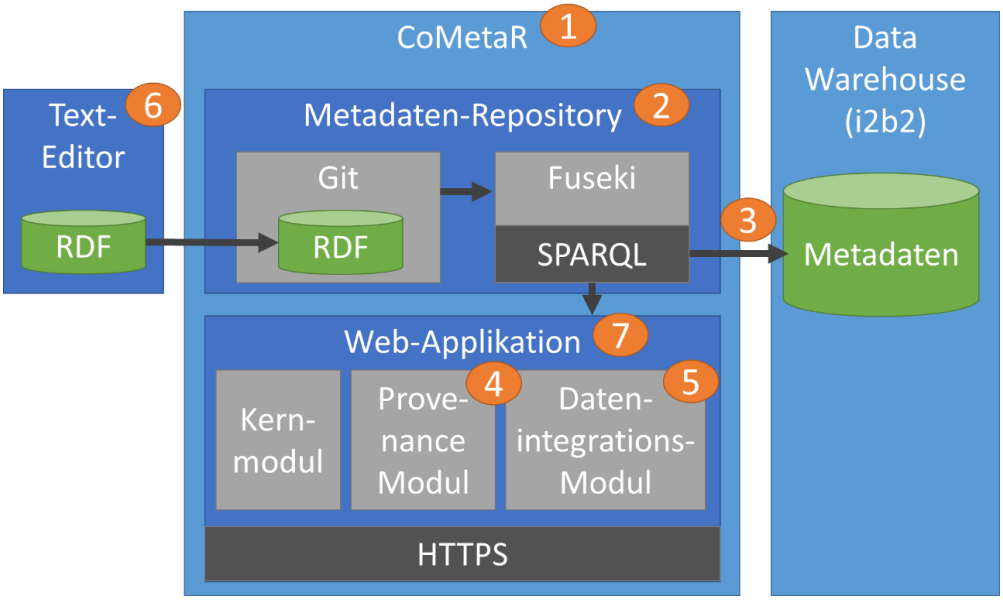


Abbildung 2: Schematischer Aufbau der Metadaten-Management-Infrastruktur. Hellblauer Kasten: Server; Dunkelblauer Kasten: Anwendung; Hellgrauer Kasten: Anwendungskomponente; Dunkelgrauer Kasten: Anwendungsschnittstelle; Grüner Zylinder: Persistente Datensammlung; Rote Kreise: Verweise zu den Publikationen von Mark Stöhr.

Das erste Manuskript liefert eine Übersicht über den Funktionsumfang von CoMetaR sowie eingesetzte Standards zur Erfassung und Kommunikation von Metadaten. Das folgende Manuskript beschreibt, wie Metadaten formal verfasst, gespeichert und bereitgestellt werden. In Publikation Nummer drei wird gezeigt, wie und in welchem Umfang diese Metadaten dann in das Metadaten-Schema der i2b2 Data Warehouse Software übertragen werden können. Im sechsten Manuskript wird ein Tool vorgestellt, das die Bearbeitung der Metadaten erleichtert. Die Publikationen vier und fünf stellen zwei Nutzerrollen-spezifische Ansichten von Metadaten innerhalb der Webapplikation vor. Die siebte Publikation stellt eine Evaluation der gesamten Web-Applikation dar und legt den Fokus auf Messung der Funktionalität und Usability.

Zunächst wurde im Beitrag *CoMetaR: A Collaborative Metadata Repository for Biomedical Research Networks* eine Übersicht zu CoMetaR und der Zielsetzung, ein Werkzeug zur Entwicklung umfassender Metadatenkataloge zu entwickeln, präsentiert. Anfangs waren die wichtigsten Funktionen die Bearbeitung und Versionierung von Metadaten, die Darstellung der Terminologie mitsamt aller Annotationen sowie eine Durchsuchbarkeit ebendieser (Fragestellung 4). Dieser Funktionsumfang der hierzu

entwickelten Web Applikation wurde später als Kern-Modul bezeichnet. Das Herzstück dieses Werkzeugs bildet das Repository selbst. Hier werden die Informationen zu allen definierten Parametern und ihrer Annotationen gespeichert und zur Verfügung gestellt. Im entsprechenden Manuskript *Using RDF and Git to Realize a Collaborative Metadata Repository* wird die Lösung zum Einsatz des Versionierungstools Git (Fragestellung 1b) in Zusammenhang mit dem Standard zur Beschreibung von Resourcen RDF (Fragestellung 1a) begründet. Zur Bereitstellung der Metadaten werden diese in einen Apache Jena Fuseki Server geladen, welcher die eigens für Kommunikation für RDF-Daten entwickelten Schnittstelle SPARQL implementiert (Fragestellung 1c).

Ein Anwendungszweck einer biomedizinischen Terminologie ist der Einsatz in einem Data Warehouse. Wie sich herausstellte, ist es prinzipiell möglich, die Definitionen von (klinischen) Parametern inklusive Struktur und wesentlicher Annotationen ins Data Warehouse zu übernehmen und dort für Bestandsabfragen zu verwenden. Im Manuskript *Metadata Import from RDF to i2b2* wird ein Algorithmus für die Überführung in RDF formulierter Metadaten in das i2b2-eigene SQL-Schema vorgestellt. Möglich war die Übernahme von Informationen zur hierarchischen Struktur, Bezeichnungen, Einheiten, Kodierungen und Beschreibungen (Fragestellung 2). Der Zusammenhang zwischen Metadaten-Repository und Data Warehouse kommt bei der Datenintegration patientenbezogener Daten zum Tragen. Um die Parameter einer Quell-Datensammlung in das zentrale Data Warehouse zu integrieren, wird ein Mapping vorausgesetzt, das die korrekte Interpretation des Parameters erlaubt. Im Falle des DZL geschieht das Mapping auf Basis eines Codes (beispielsweise eines LOINC Codes), der die Bedeutung klar festlegt. Diese Mappings sind – wie im Manuskript *Verifying Data Integration Configurations for Semantical Correctness and Completeness* gezeigt – technisch formuliert und für weniger technisch versiertes Personal unverständlich. Es wurde deshalb eine Möglichkeit entwickelt, diese Mappings, welche in Konfigurationsdateien zusammengefasst für den Datenupload ins Data Warehouse verwendet werden, in unserer Web Applikation zu analysieren und visuell darzustellen (Fragestellung 5). Das entsprechende Modul wird Datenintegrations-Modul genannt. Unser Ansatz zeichnet sich vor allem dadurch aus, dass er den annotierten Ziel-Katalog zeigt und die Quell-Parameter zuordnet. Dadurch bekommen Dateninhaber und -erfasser, welche die Quell-Parameter für gewöhnlich sehr gut kennen, alle nötigen Informationen übersichtlich und zusammenhängend präsentiert.

Neben der Überprüfung von Mappings auf Korrektheit und Vollständigkeit ist für einen Daten liefernden Standort aber auch von Bedeutung, welche Änderungen am Metadatenkatalog vorgenommen wurden. Nur in Kenntnis dieser Änderungen können nötige Anpassungen an den Konfigurationsdateien vorgenommen werden. Die Information über Änderungen des Metadatenkatalogs sind aber auch für diejenigen von großem Interesse, welche einen speziellen Bereich oder die gesamte Terminologie überblicken und ihre semantische Konsistenz wahren. Um dieser Notwendigkeit gerecht zu werden, wurde das Provenance-Modul entwickelt, welches im Manuskript *Provenance for Biomedical Ontologies with RDF and Git* vorgestellt wird (Fragestellung 6). Der große Vorteil dieser Lösung ist, dass nahezu alle relevanten Informationen aus dem Versionierungstool Git extrahiert und somit auch nachträglich bestimmt werden können.

Um sicherzustellen, dass die zuvor beschriebenen Module (Kern-Modul, Datenintegrations-Modul, Provenance-Modul) den Anforderungen hinsichtlich Funktionalität und Nutzerfreundlichkeit erfüllen, wurde eine umfangreiche Evaluation durchgeführt. Zwölf Mitarbeiter, welche in den Datenintegrations-Prozess des DZL involviert sind, haben die für ihre Tätigkeit relevanten Module genutzt und Fragebögen ausgefüllt, die die Nutzerfreundlichkeit messen. Methodik und Ergebnisse sind im Manuskript *The Collaborative Metadata Repository (CoMetaR) Web Application: Quantitative and Qualitative Usability Evaluation* nachzulesen und bergen neben den grundsätzlich positiven Ergebnissen auch Potenzial für Verbesserungen. Diese Arbeit ist in Zusammenhang mit den Fragestellungen 4, 5 und 6 zu sehen, da diese nur als beantwortet betrachtet werden können, wenn die entsprechenden Informationen auch nutzerfreundlich präsentiert werden.

Im Laufe der Zeit hat sich herausgestellt, dass die Bearbeitung des Metadatenkatalogs in reinen Textdateien vielen Mitarbeitern Schwierigkeiten bereitet. Beispielweise werden syntaktische Fehler erst beim Hochladen zurückgemeldet. Genauso sind Änderungen an der hierarchischen Struktur in reiner Textform schwer nachzuvollziehen. Im Zuge dessen wurde eine Erweiterung für einen weit verbreiteten Texteditor entwickelt, siehe Manuskript *ISO 21526 Conform Metadata Editor for FAIR Unicode SKOS Thesauri*. Ziel dieser öffentlich zugänglichen Erweiterung ist ein erleichterter Zugang zur Teilnahme am Erarbeitungsprozess unseres Metadatenkatalogs (Fragestellung 3).

4. Manuskripte

MEDINFO 2017: Precision Healthcare through Informatics
A.V. Gundlapalli et al. (Eds.)

doi:10.3233/978-1-61499-830-3-1337

CoMetaR: A Collaborative Metadata Repository for Biomedical Research Networks

Mark R. Stöhr, Gudrun Helm, Raphael W. Majeed, Andreas Günther

UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany

Abstract

The German Center for Lung Research (DZL) is a research network with the aim of researching respiratory diseases. To perform consortium-wide queries through one single interface, it requires a uniform conceptual structure. No single terminology covers all our concepts. To achieve a broadly accepted and complete ontology, we developed a platform for collaborative metadata management "CoMetaR". Anyone can browse and discuss the ontology while editing can be performed by authenticated users.

Keywords:

Metadata; Terminology; Intersectoral Collaboration

Introduction

The DZL is a consortium of multiple lung research institutions. We collectively pursue the goal to find ways of preventing and curing respiratory diseases. There are many different local data storing systems in use, e.g. CentraXX, Filemaker etc. This circumstance hinders researchers from performing consortium-wide queries quickly and with least effort. Therefore, a central data warehouse (i2b2) is used to which every site uploads their data. Semantic integration of lung research data requires not just one but multiple existing terminologies like LOINC and SNOMED-CT, in addition to custom lung research specific concepts. The OBO Foundry lists four principles for developing a new ontology [1]: (i) be developed in a collaborative effort, (ii) use common relations that are unambiguously defined, (iii) provide procedures for user feedback and for identifying successive versions and (iv) have a clearly bounded subject-matter. Aim of this project is to realize a platform that visualizes the DZL metadata ontology and enables medical documentalists and researchers from all participating institutions to take part in the development process.

Methods

Our requirement analysis resulted in the following statements: The ontology has to be visualized and searchable through an user interface, which is accessible by any person without additional software. Medical documentalists need to maintain the ontology, which should be possible without additional software. The description format has to be extendable. Any DZL member should be able to contribute expertise. Every term of the ontology needs to be discussable and discussion history itself accessable. Trained specialists from anywhere should be able to take part in editing the ontology independently and simultaneously. Changes to the ontology need verification, e.g. every concept needs to be labeled and properly integrated into the hierarchy. Investigation of existing solutions [2,3,4] did not lead to satisfying results.

Results

We developed a platform for independently maintaining an ontology from different sites combined with a web interface for visualization and concept related discussion. The web interface is based on standard technologies in order to guarantee accessibility. All metadata concepts can either be explored through an expandable tree or a search form. For each concept the user is offered details and a discussion board. RDF was chosen as ontology description format because of it's basic purpose for graph description, it's extensibility as well as the possibility of editing in simple text editor. Concept hierarchy is realized through the Simple Knowledge Organization System (SKOS) relations "broader" and "narrower". All RDF files are stored in a GIT repository, to which any person with access can upload data. Transmissions are verified through syntactical and semantical tests. Afterwards, the updated ontology is immediately loaded into the i2b2 server and a triple-store. The latter provides all information for the web interface via AJAX and standard SPARQL queries.

Discussion

During development we focused on using existing standards. CoMetaR may be used for arbitrary SKOS ontologies. It is especially suitable for management of biomedical and other evolving ontologies. Additional functionality may be integrated into the web interface through javascript extensions.

Conclusions

We developed a software for collaborative management of metadata ontologies. Specialists from all participating institutions are able to view and contribute. All source code is open source and available at https://github.com/dzl-dm/cometar .

References

[1] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L.J. Goldberg, K. Eilbeck, A. Ireland, C.J. Mungall et al., The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration., *Nat Biotechnol* **25** (2007), 1251-1255.
[2] Protégé, http://protege.stanford.edu, Accessed 4/17.
[3] BioPortal, http://bioportal.bioontology.org, Accessed 4/17.
[4] OBO-Edit, http://oboedit.org/, Accessed 4/17.

Address for correspondence

Mark R. Stöhr, e-mail: mark.stoehr@innere.med.uni-giessen.de

Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth
A. Ugon et al. (Eds.)

doi:10.3233/978-1-61499-852-5-556

Using RDF and Git to Realize a Collaborative Metadata Repository

Mark R. STÖHR[1], Raphael W. MAJEED and Andreas GÜNTHER
UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany

Abstract. The German Center for Lung Research (DZL) is a research network with the aim of researching respiratory diseases. The participating study sites' register data differs in terms of software and coding system as well as data field coverage. To perform meaningful consortium-wide queries through one single interface, a uniform conceptual structure is required covering the DZL common data elements. No single existing terminology includes all our concepts. Potential candidates such as LOINC and SNOMED only cover specific subject areas or are not granular enough for our needs. To achieve a broadly accepted and complete ontology, we developed a platform for collaborative metadata management. The DZL data management group formulated detailed requirements regarding the metadata repository and the user interfaces for metadata editing. Our solution builds upon existing standard technologies allowing us to meet those requirements. Its key parts are RDF and the distributed version control system Git. We developed a software system to publish updated metadata automatically and immediately after performing validation tests for completeness and consistency.

Keywords. Biological ontologies, metadata, common data elements, data pooling, organization and administration

1. Introduction

The DZL is a consortium of multiple lung research institutions. We pursue the goal to find ways of preventing and curing respiratory diseases. The DZL divides into several disease areas and registers collecting differing data depending on their studies' focus. From a technical point of view, we are confronted with a variety of historical grown, site-specific software systems like Excel, Access, CentraXX, Filemaker, SecuTrial, etc. This circumstance hinders researchers from performing consortium-wide queries. That is why we use a central data warehouse (i2b2) to collect data from all local databases and information systems, offering one single interface to query all data. In the first stage, we focus on the DZL's data elements that are common to multiple data sets across different studies, so-called common data elements (CDE) [1]. An example would be the CDE "body height", represented in many registries, varying in unit (cm/m/ft/..), value type (float/integer) and description ('height'/ 'size'/..). Collecting these CDE will allow us to do retrospective research on big data pools. Beforehand, we need to understand and interpret received data in order to convert matching CDE's representations into one harmonized representation. To define such collection of harmonized data representations,

[1] Corresponding author: mark.stoehr@innere.med.uni-giessen.de.

we depend on metadata: data describing data elements in their specific context [2]. Hence, for semantic integration, we require a lung research metadata collection. Existing terminologies like Logical Observation Identifiers Names and Codes (LOINC) [3] and Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [4] do not provide sufficient granularity and lack coverage of many lung research specific concepts. To develop lung research specific metadata, we need a central metadata repository / metadata registry (MDR): a storage of metadata allowing editors to add and modify concepts [2,5]. Research has shown the importance of teamwork in the process of creating metadata repositories [6] - not only for growing expertise but also in terms of willingness to participate in future steps of data integration. This paper is about developing a repository allowing members to create the DZL metadata collaboratively. Existing metadata management software systems show the feasibility of this approach [5, Löbe], e.g. caDSR (Cancer Data Standards Repository) [7], the UK CancerGrid [8], Ontobrowser [9], Webprotégé [10] and the Samply MDR [11].

2. Methods

2.1. Requirements

For defining our metadata repository's requirements, we involved 18 data managers from all DZL participating sites. After several conferences, we concluded the following needs:

The key part of a metadata repository is the **representation (i)** of metadata. We need to depict a concept's labeling and hierarchical classification, e.g. tumor stage T1a is a subclass among others within the tumor stage T1. **Usability for editors (ii)** is required to minimize the effort of adding and modifying metadata. Since our approach is to develop an ontology that fulfills needs of all participating study sites, simultaneous editing and committing is required – resulting in the need for **editing conflict management (iii)**. For tracking changes and always being able to provide a consistent state, we require **version control (iv)**. Changes committed to the metadata should be automatically checked for correctness through **content verification (v)**, e.g. detection of missing labels or contradictions. We need **system simplicity (vi)** to minimize the effort to maintain and understand the repository's software solution. To guarantee project continuity after and between funding periods, we require **low costs (vii)**. **Standard conformity (viii)** provides compatibility to other systems building on same standards and also allows the research community to benefit from our developments. To cope with future developments and changing research demands, we require **extensibility (ix)**. Finally yet importantly, **access control routines (x)** must be implemented.

2.2. Satisfying software/standards

We evaluated the previously mentioned existing software systems. They did not meet our requirements, mostly because of low usability, complicated software installation and insufficient metadata representation capabilities. Nevertheless, by considering their methods and evaluating established technologies, we found a suitable solution:

The Resource Description Framework (**RDF**) [12] is a formal language for representation of ontologies (i). It is a cross-domain standard providing flexibility in terms of notation and extension (ix). If notated in Turtle syntax [13], it consists of short human-readable paragraphs (ii). This allows extension by simply appending triples to

existing statements. **Git** [14] is an open source distributed version control system. By itself it provides conflict management (iii), version control (iv), parallel offline content editing and even a graphical user interface (ii). **Apache Jena Fuseki** [15] is an open source RDF framework for semantic networks. Fuseki provides a SPARQL Protocol And RDF Query Language (**SPARQL**) [16] interface, which makes all data stored in the Jena server accessible via queries. By that, it allows us to perform extensive testing (v) on the present metadata. **Apache Web Server** provides access control (x) to Git repositories over the http-protocol. All mentioned software solutions are open source (xii) and common standard (xiii). By connecting them, we gain a transparent system (vi).

3. Results

Our Metadata Repository consists of three main applications hosted on a Debian 8 Linux server (figure 1). The Apache Webserver provides external access and access control. RDF data is sent by Git clients via https protocol and forwarded to a Git repository. The Git repository serves as a persistent data storage including versioning. Data loaded into Jena Fuseki server always mirrors the data from Git repository with hierarchical inference rules applied (e.g. broader vs. narrower, topConceptOf vs. hasTopConcept).

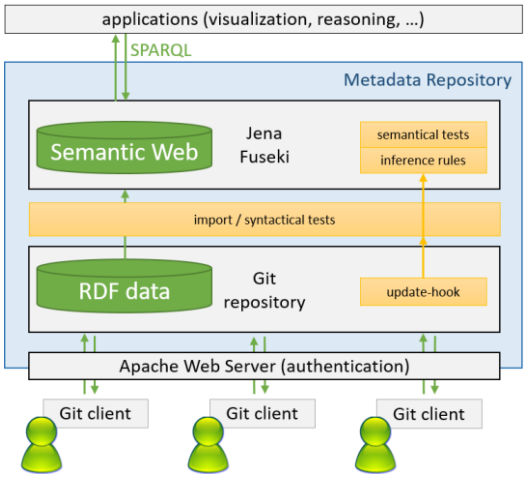


Figure 1. Metadata repository's technical infrastructure

Git allows every editor to keep a local copy of all files in the repository. The DZL ontology is stored as plain text in Turtle syntax. Every concept is described in its own paragraph consisting of several statements (figure 2). A statement consists of three parts: subject, predicate and object. The predicate stands for a relation between subject and object, where object can be a literal or another concept. One may shorten consecutive statements referring to the same subject by using a semicolon. For assigning attributes and relations, we use the Simple Knowledge Organization System (SKOS) [17]. One may assign labels in any language as well as hierarchical relations to other concepts, e.g. "BloodDeriv" is sub-concept of "Specimen", indicated by the relation "broader". We also implemented the attribute "skos:notation", which lets us annotate codes from coding systems like LOINC and SNOMED-CT. The attribute "skos:topConceptOf" marks the root of a concept tree. After editing, a user can commit changes as follows: (1) Mark

changes that should be committed. (2) Merge own changes with changes from other editors. (3) Describe changes with a commit message. (4) Push data to the server.

Figure 2. Excerpt of an RDF file written in Turtle syntax

Git allows us to execute scripts ("hooks") like the *update*-hook, which takes effect after receiving the data and just before storing it in the repository. By then, our software is executed doing the following: (1) Check whether the new data can be successfully loaded into a temporary Jena Fuseki instance. If so, the RDF statements were syntactically correct. (2) Insertion of inference rules like "A broader B => B narrower A". (3) Perform consistency checks on the data. We require all concepts to have exactly one preferred English label and that assigned codes are unique. We also verify that all concepts have a hierarchical path to a root concept. (4) In the last step, the verified data is exported from the temporary Jena Fuseki server and loaded into our Jena Fuseki live server. Since we perform all verifications on a temporary instance before publishing, the accessible data is always in a consistent state. Nevertheless, our system immediately publishes the updated metadata after an editor's data submission.

Our ontology consists of 692 items, notated in 40 Turtle-files. 178 items are annotated with codes from other ontologies (135 SNOMED-codes, 33 LOINC-codes, 10 SPREC 2.0-codes, 54 ICD-O-codes). In total, four people (three data managers and one clinical documentalist) contributed to these files by following the mentioned steps.

4. Discussion

Our solution of a metadata repository is composed of standard technologies. It is built modularly to allow easy adaption to custom environments. For instance, the Apache Web Server as authentication layer may be left out or replaced with other authentication services like htaccess, OAuth or LDAP/Active directory. Currently, the metadata is edited in a simple text editor. This may lead to difficulties for technically less versed users, because such editors do not prevent potential syntax errors and lack visual clarity. The editing process could be improved through graphical enhanced RDF editing software (e.g. Protégé by Stanford University). Instead of applying inference rules, one may integrate a reasoner into the triple store, which was not necessary for our needs. The standard SPARQL interface provides connectivity for additional software applications like third-party visualization and reasoning tools. For example, we developed an RDF

ontology visualization tool for displaying and browsing the concept tree. It is also planned to include a discussion board to support communication for editors and reviewers on concept level. Both, the metadata repository and the concept tree browser are part of the Collaborative Metadata Repository (CoMetaR) project. It is open source and hosted on GitHub (https://github.com/dzl-dm/cometar).

5. Conclusion

We successfully developed a metadata repository serving our requirements. It follows our collaborative approach, builds on standard technology, is easy to use, flexible and maintainable. The setup allows all DZL members to contribute to the developing process of our DZL central data warehouse metadata. Future work includes further mapping to common ontologies, user evaluation and enabling concept-related discussion.

6. Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] National Institutes of Health, Common Data Element (CDE) Resource Portal, 2016 [Last accessed: 11/10/2017], https://www.nlm.nih.gov/cde/.
[2] International Standards Organization, ISO/IEC 111179, Information Technology – Metadata Registries (MDR), 2004 [Last accessed: 11/10/2017], http://metadata-stds.org/11179/.
[3] Regenstrief Institute, LOINC home page, 2016 [Last accessed: 11/10/2017], http://www.regenstrief.org/resources/loinc/.
[4] College of American Pathologists, SNOMED Clinical Terms (SNOMED CT), 2017 [Last accessed: 11/10/2017], http://www.snomed.org.
[5] Matthias Löbe, *IT-Infrastrukturen in der patientenorientierten Forschung*, TMF – Technologie- und Methodenplattform für die vernetzte medizinische Forschung e. V., Berlin, Germany, 2015.
[6] Marlene Z. Cohen, Implementing Common Data Elements Across Studies to Advance Research, *Nurs Outlook* **63(2)** (2015), 181–188.
[7] Nadkarni, P.M., Brandt, C.A., The Common Data Elements for cancer research: remarks on functions and structure, *Methods Inf Med* **45(6)** (2006), 594-601.
[8] Papatheodorou, I., Crichton, C., Morris, L. et al., A metadata approach for clinical data management in translational genomics studies in breast cancer, *BMC Med Genomics* **2** (2009), 66.
[9] C. Ravagli, OntoBrowser: a collaborative tool for curation of ontologies by subject matter experts., *Bioinformatics* **33(1)** (2017), 148-149
[10] Tudorache, Tania, Jennifer Vendetti, and Natalya Fridman Noy, Web-Protege: A Lightweight OWL Ontology Editor for the Web, *OWLED* **432** (2008).
[11] Samply MDR, The Metadata Repository (MDR), 2012 [Last accessed: 11/10/2017], https://mdr.ccpit.dktk.dkfz.de/index.xhtml.
[12] W3C Semantic Web, Resource Description Framework (RDF), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/rdf.
[13] W3C Semantic Web, Turtle – Terse RDF Triple Language, 2014 [Last accessed: 11/10/2017], https://www.w3.org/TR/turtle/.
[14] Git --local-branching-on-the-cheap, 2017 [Last accessed: 11/10/2017], http://git-scm.com/.
[15] Apache Jena Fuseki, 2017 [Last accessed: 11/10/2017], http://jena.apache.org/documentation/fuseki2/.
[16] W3C Semantic Web, SPARQL Protocol And RDF Query Language (SPARQL), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/sparql.
[17] W3C Semantic Web, Simple Knowledge Organization System (SKOS), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/skos.

German Medical Data Sciences: A Learning Healthcare System
U. Hübner et al. (Eds.)

doi:10.3233/978-1-61499-896-9-40

Metadata Import from RDF to i2b2

Mark R. STÖHR[a,1], Raphael W. MAJEED[a] and Andreas GÜNTHER[a]
[a] *UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany*

Abstract. Metadata management is an important task in medical informatics and highly affects the gain out of existing health information data. Data Warehouse solutions like Informatics for Integrating Biology and the Bedside (i2b2) are common tools for identifying patient cohorts and analyzing collected clinical data while respecting patient privacy. The Resource Description Framework (RDF) is designed for highly interoperable ontology representation in various formats, facilitating ontology and metadata management. Our approach is to combine i2b2's and RDF's benefits by importing the easy-to-edit RDF ontology into the extensive-research-enabling i2b2 software. We do so by using a SPARQL Protocol and RDF Query Language (SPARQL) interface, that enables RDF data queries, and developing a java program, which then generates i2b2-specific SQL insert statements. To demonstrate our solution's feasibility, we transcribe our lung disease specific ontology to RDF and import it into our i2b2 data warehouse.

Keywords. Biological ontologies, metadata, organization and administration, automatic data processing, information systems

1. Introduction

In medical informatics, metadata management is an important and demanding task. It is indispensable for data harmonization, data integration, data quality management and data comparison. Applied on clinical data, these processes result in large pools of consolidated, corrected and annotated data, enabling large-scale clinical data analysis and trial patient recruitment [1].

The clinical data warehouse software i2b2 serves as storage system for clinical data as well as metadata and offers a reliant and effective tool to support clinical trials by either prospectively finding cohorts of patients fulfilling specific constraints or retrospectively making further use of already collected routine health care data. Established in 2004, 7 years later already 60 academic health care centers have adopted this software [2]. It is funded by the National Institutes of Health (NIH). A large community continuously enhances its features. For example, by developing solutions for challenges created by i2b2's founders at Harvard MIT Division of Health Sciences and Technology. From 2006 to 2012 there are 124 publications listed, enabled by these challenges [3]. From 2016/01 to 2017/10 PubMed lists 46 articles related to i2b2, which shows that i2b2 and its applications are of ongoing interest and usefulness to the research community.

[1] Corresponding Author, Mark R. Stöhr, UGMLC, Justus-Liebig-University, Klinikstraße 36, 35392 Gießen, Germany; E-mail: mark.stoehr@innere.med.uni-giessen.de.

RDF is a W3C standard and designed for ontology representation and described resources may be annotated with literals (e.g. labels or codes) and connected to other resources for large knowledge-graphs. It is commonly used in many fields like content management, content discovery, data integration, semantic annotation and schema mapping. Several use cases show Health Care applications' clear dependency on these fields [4]. RDF statements consist of triples < Subject, Predicate, Object >, which describe the subject in more detail, either through literal attributes or through relations to other resources. RDF has a variety of representations, e.g. N3, Turtle, JSON or XML [5–7].

For research (meta)data, FAIR Data Principles offer a measurable set of principles to improve data quality in terms of data being Findable, Accessible, Interoperable and Reusable. Many implementations show how RDF serves as a reliant component for fulfilling these principles [8]. One may enrich RDF resources by referencing concepts from clinical ontologies like SNOMED-CT or LOINC through their globally unique identifier / code [9,10]. For this purpose, systems like Dublin Core (DC) or Simple Knowledge Organization System (SKOS) provide standardized relations and annotations [11,12]. By referencing common resources in a standardized way, RDF data becomes interoperable, findable and reusable.

Although several solutions exist for importing routine health care data [13], there are still very few tools for i2b2 ontology administration [14]. Particularly, there is no published solution for importing RDF metadata into i2b2.

This paper aims on developing a generic solution for metadata import of RDF metadata into an i2b2 database and prove its feasibility.

2. Methods

There are two effective ways for importing data into the i2b2 metadata structure: Either by working directly on the database or by using the provided web service API. The latter becomes less efficient for large-scale data import and is less flexible because of API restrictions. Therefore, we decided to query the database directly through SQL statements. To achieve this, we have to transform RDF resources into appropriate SQL statements, taking into account their relations (especially their hierarchy) as well as some annotations like labels and codes of other ontologies.

Four database tables are used for the i2b2 metadata representation: "i2b2", "table_access", "concept_dimension" and "modifier_dimension" [15]. The tables "i2b2" and "table_access" contain information about the visual representation of concepts in the user interface. This is where labels and descriptions are stored. The "concept_dimension" and "modifier_dimension" tables contain coding information. They are used internally for execution of user queries. Modifiers are implemented in i2b2 since release 1.6.00 to enrich existing concepts (and when needed its sub-concepts) with additional specification options. I2b2 can only display concept trees, so we can only import RDF graphs without loops.

The W3 Consortium published SPARQL [16] as a standard for querying RDF. By using an engine with a SPARQL-interface, we can provide accessibility (see FAIR principles) of RDF resources. Jena Fuseki Server is a popular open source software designed for querying loaded RDF data via SPARQL interface. It supports a wide range of input formats like Turtle, XML or JSON.

For each concept or modifier, we have to query the RDF data for the concept's primary label, codes, as well as sub-concepts and applied modifiers. Then we add a row to the i2b2 table. In case of annotated codes, which are necessary for concepts and modifiers to be found and considered in the results, we also add a row to the respective "*_dimension" table. In case of a root concept, we also add a row to the "table_access" table.

Knowing how to query the RDF data and what the i2b2 database insert statements have to look like, we are able to formulate an algorithm that will generate all insert statements by successively gathering all required concept and modifier information from the RDF graph (Figure 1). This algorithm recursively runs through a method that, for each concept, queries all its sub-concepts, modifiers, notations and its preferred label, generates the proper SQL-statement and then continues with all sub-concepts. Visual attributes for the i2b2 web client (e.g. folder, leaf, hidden sub-concepts) are deduced by looking at the number of sub-concepts. For modifiers, the same procedure is applied.

The German Center for Lung Research (DZL) uses i2b2 as central data warehouse software, collecting data from various sources and merging them for large-scale retrospective data analysis, sample ordering and patient recruitment. The advantages of properly setup metadata has already been shown in the introduction. To evaluate our metadata import algorithm, we transcribe our DZL ontology to turtle N3 syntax and load it into a Jena Fuseki server, offering the SPARQL interface for querying the data. The DZL ontology was composed by lung researchers, medical documentalists and data managers using our Collaborative Metadata Repository (CoMetaR) editing framework [17].

For hierarchical ordering of concepts and modifiers we use the standard RDF predicates skos:topConceptOf/skos:hasTopConcept and skos:broader/skos:narrower. For modifier indication, we use rdf:hasPart/rdf:partOf. For labeling and description, we use skos:prefLabel and dc:description. Concept's and modifier's codes are indicated through skos:notation. Table 1 shows an example of entries in the postgres database used by i2b2. All software components (RDF files, Jena Fuseki Server, our java program and i2b2 software) run on the same Linux system.

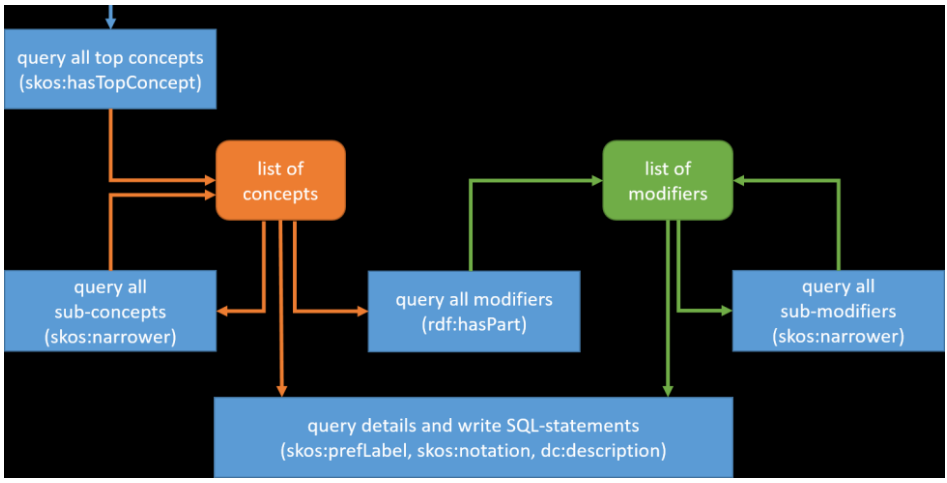


Figure 1. Visualization of the RDF-to-i2b2 transformation algorithm.

Table 1. Example entries in respective database tables for i2b2 (postgres).

Database table i2b2

c_fullname	c_name	m_applied_path
\i2b2\dzl:Dataset\dzl:Follow-up\	Follow-up	NULL
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\	Vital sign	NULL
\L:8462-4\	Diastolic blood pressure	\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\%
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\	Blood pressure	NULL

Database table concept_dimension

concept_path	name_char	concept_cd
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\	Blood pressure	S:75367002

Database table modifier_dimension

modifier_path	name_char	modifier_cd
\L:8462-4\	Diastolic blood pressure	L:8462-4

3. Results

We developed a solution for integrating RDF ontologies into i2b2 databases by loading the data into an application that provides a SPARQL interface processing it through an algorithm, which queries the interface for RDF concepts recursively and generates SQL insert statements, before executing these SQL-statements. The ontology is accessible via SPARQL through https://data.dzl.de/fuseki/cometar_live/query. A visualization is available at https://data.dzl.de/cometar/.

3.1. Range and Performance of the Implementation

Our entire DZL ontology was successfully imported into i2b2. Currently it includes 653 concepts and 39 modifiers organized in four trees with 518 leafs. Out of all, 601 concepts and 32 modifiers are associated with a code. In total, the ontology is spread across 40 text files in turtle syntax (ttl). During transformation by our java algorithm, all insert statements are split up into two SQL files, since some have to be executed on i2b2metadata schema and some on i2b2demodata schema. Loading the ttl-files into Jena Fuseki Server takes less than one second. The java program for generating all i2b2 insert statements runs between 9 and 21 seconds. Executing the SQL files on PostgreSQL takes about two seconds.

4. Discussion

We developed a generic algorithm to import metadata from RDF to i2b2 and proved its feasibility. Although i2b2 can represent some basic elements of standard RDF vocabulary (e.g. SKOS), its capabilities are rather limited. For example, relations like skos:closeMatch and graphs with loops are not representable in i2b2. On the other hand, data type restrictions are supported, but our algorithm does not yet consider them. Editing

RDF ontologies in turtle N3 syntax by hand may prove difficult. In this case, RDF editing tools like Protégé may facilitate this task. Since execution time of our implementation lies under one minute, it is possible to perform live i2b2 ontology updates after editing the RDF ontology.

5. Conclusion

In order to combine the benefits of i2b2 and RDF, we investigated possibilities to transfer RDF metadata to i2b2 data warehouse systems. We found that import of hierarchical information, labels and codes is feasible and developed an algorithm for this task.

6. Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] Q. Chong, A. Marwadi, K. Supekar, Y. Lee, Ontology Based Metadata Management in Medical Domains, *Journal of Research and Practice in Information Technology* **35(2)** (2003), 139–154.
[2] I.S. Kohane, S.E. Churchill, S.N. Murphy, A translational engine at the national scale: informatics for integrating biology and the bedside, *Journal of the American Medical Informatics Association : JAMIA* **19(2)** (2012), 181–185.
[3] Informatics for Integrating Biology and the Bedside, NLP Research Data Sets, 2017 [Last accessed: 10/27/2017], https://www.i2b2.org/NLP/DataSets/Publications.php.
[4] W3C Semantic Web, Semantic Web Case Studies and Use Cases, 2012 [Last accessed: 11/10/2017], https://www.w3.org/2001/sw/sweo/public/UseCases/.
[5] W3C Semantic Web, W3C Recommendation / RDF Documents and Syntaxes, 2014 [Last accessed: 11/10/2017], https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html.
[6] W3C Semantic Web, Notation3 (N3): A readable RDF syntax, 2011 [Last accessed: 11/10/2017], https://www.w3.org/TeamSubmission/n3/.
[7] W3C Semantic Web, Turtle – Terse RDF Triple Language, 2014 [Last accessed: 11/10/2017], https://www.w3.org/TR/turtle/.
[8] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3** (2016), 160018.
[9] Regenstrief Institute, LOINC home page, 2016 [Last accessed: 11/10/2017], http://www.regenstrief.org/resources/loinc/.
[10] College of American Pathologists, SNOMED Clinical Terms (SNOMED CT), 2017 [Last accessed: 11/10/2017], http://www.snomed.org.
[11] Dublin Core Metadata Initiative, Dublin Core, 2017 [Last accessed: 11/10/2017], http://dublincore.org/.
[12] W3C Semantic Web, Simple Knowledge Organization System (SKOS), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/skos.
[13] R.W. Majeed, R. Röhrig, Automated realtime data import for the i2b2 clinical data warehouse: introducing the HL7 ETL cell, *Studies in health technology and informatics* **180** (2012), 270–274.
[14] C.R. Bauer, T. Ganslandt, B. Baum, et al., Integrated Data Repository Toolkit (IDRT). A Suite of Programs to Facilitate Health Analytics on Heterogeneous Medical Data, *Methods of information in medicine* **55(2)** (2016), 125–135.
[15] Informatics for Integrating Biology and the Bedside, i2b2 software, 2017 [Last accessed: 11/10/2017], https://www.i2b2.org/software/index.html.
[16] W3C Semantic Web, SPARQL Protocol And RDF Query Language (SPARQL), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/sparql.
[17] M.R. Stöhr, R.W. Majeed, A. Günther, Using RDF and Git to Realize a Collaborative Metadata Repository, *Studies in health technology and informatics* **247** (2018), 556–560.

German Medical Data Sciences: Shaping Change – Creative Solutions for Innovative Medicine
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI190832

Provenance for Biomedical Ontologies with RDF and Git

Mark R. STÖHR [a,1], Andreas GÜNTHER [a] and Raphael W. MAJEED [a]
[a] *UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany*

Abstract. The German Center for Lung Research (DZL) is a research network with the aim of researching respiratory diseases. In order to enable consortium-wide retrospective research and prospective patient recruitment, we perform data integration into a central data warehouse. The enhancements of the underlying ontology is an ongoing process for which we developed the Collaborative Metadata Repository (CoMetaR) tool. Its technical infrastructure is based on the Resource Description Framework (RDF) for ontology representation and the distributed version control system Git for storage and versioning. Ontology development involves a considerable amount of data curation. Data provenance improves its feasibility and quality. Especially in collaborative metadata development, a comprehensive annotation about "who contributed what, when and why" is essential. Although RDF and Git versioning repositories are commonly used, no existing solution captures metadata provenance information in sufficient detail. We propose an enhanced composition of standardized RDF statements for detailed provenance representation. Additionally, we developed an algorithm that extracts and translates provenance data from the repository into the proposed RDF statements.

Keywords. Biological ontologies, metadata, data curation, automatic data processing, quality improvement.

1. Introduction

1.1. Background

The German Center for Lung Research (DZL) is a consortium of multiple lung research institutions. We pursue the goal to find ways of preventing and curing respiratory diseases. The DZL divides into several disease areas collecting differing data depending on their studies' focus. From a technical point of view, we are confronted with a variety of historically grown site-specific software systems like Excel, Access, CentraXX, Filemaker, SecuTrial, etc. This circumstance hinders researchers from accessing the complete consortium-wide data inventory. That is why we use a central data warehouse (i2b2) to store data from all local databases and information systems, offering one single interface to query all patient related data. The underlying metadata ontology covers concepts related to the lung research domain. Our metadata development is ongoing and technically supported by our Collaborative Metadata

[1] Corresponding Author, Mark R. Stöhr, UGMLC, Justus-Liebig-University, Klinikstraße 36, 35392 Gießen, Germany, E-Mail: mark.stoehr@innere.med.uni-giessen.de

Repository (CoMetaR) tool. It is based on the Resource Description Framework (RDF) for ontology representation and the distributed version control system "Git" for version control [1,2].

We designed the system to satisfy the FAIR principles [3]. Although we assess the principles "Findable", "Accessible" and "Interoperable" as fulfilled, it still misses an important aspect of "Reusability": Provenance, a record of information that enables researchers to track the range of, participants in and reasons for changes that were made to the ontology. The benefits of provenance are not limited to a specific domain, but can be broadly applied to any kind of data [4,5]. It has been shown to be an absolute requirement for data curation processes, since it offers the ability of tracing and reproducing changes by making them auditable, verifiable and reproducible [6,7].

Although Git is often used for knowledge resource versioning, there are only few attempts to extract provenance data.

1.2. Requirements

Through literature research and our own ongoing data curation process, we identified several use cases: (1) during the data curation process, a user needs to inquire with a concept's author and involved people for clarification. (2) For measuring the ontology's progression, we need to calculate its size and number of changes broken down into time intervals. (3) To minimize curation effort, we do not only want to record what concepts have been modified, but exactly what attributes have been added/removed/changed. (4) There are cases in which we need to change a concept's identifier. Thereby, we lose the link between the concept and its precursors, thus cannot track the concept's entire history. To solve this, the respective link needs to be documented in the system. (5) Besides the annotations introduced by Auer et al. [8], we need the information of who else was involved in specific changes. For example, a medical documentalist may act on behalf of a medical investigator. (6) All information must be extracted from the existing GIT repository. (7) The resulting provenance files should contain all relevant information while also being as compact as possible.

2. State of the art

In the field of informatics, the requirement for provenance has been identified and applied to productive systems. There are solutions to provide provenance data based on Git: The Git4Voc project makes use of "hooks", which are script interfaces provided by Git, that execute at certain points during the data upload process [9]. The Git2PROV project extracts basic information about repository versions such as the author and a description [10]. The Quit Store is a multi-layered system, that uses Git as a backend tool for versioning [11].

We found several publications for analyzing and applying provenance principles in the domain of medical informatics, e.g. by McGovern [12] and Sahoo [13]. They usually refer to provenance of clinical data analysis, study design, workflow management, etc., but none of them addresses the development of metadata itself.

Gonçalves et al. presented an algorithm for detecting and presenting changes between OWL ontologies [14]. Although they identify additions and removals down to an atomic level, they do not serialize them in a standardized way. Auer et al. present one approach to record RDF Knowledge Base data evolution on an atomic level [8].

For enhancing human change review, they annotate changes with information about what changes have been done, the editing user, timestamp, documentation and the bundle to which a change belongs. They introduce the "log" namespace, which is one of multiple frameworks dealing with the task of annotating changes of data in a standardized way. Other examples are EvoRDF [15], the Quit Store [11], Delta [16] and ChangeSet [17].

The latest standard by the World Wide Web Consortium (W3C) to record provenance data is the PROV document, including the sub-document PROV-O for ontologies [18]. This framework is increasingly applied in the international research community for medical informatics [13,19,20]. Its advantage is the adaptiveness to many fields through its generic design. By itself, it is not specific enough to describe (meta-) data changes on an atomic level, but PROV relations are designed to be qualifyable with additional information. The three main classes are Agents, Actions and Entities. In our case, we deal with an ontology containing concepts (entities) on which the changes (actions) are performed by medical staff (agents).

3. Concept

The following figure illustrates a typical workflow, in which three users simultaneously edit the RDF ontology. To gather provenance information about an ontology state and the changes made, the states' precursors have to be identified.

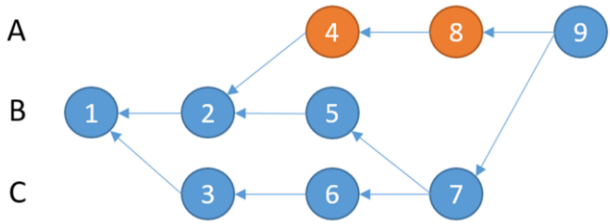


Figure 1 – Example of ontology versioning and merging. Blue circles: successfully uploaded ontology versions; red circles: failed uploads; arrows: succession.

Explanation: User B is author of ontology state 1. User C fetches this version and both users make changes to state 1, resulting in state 2 and 3. User B immediately pushes his changes to the repository and User A fetches those (4). After users B and C made changes to their local copies (5,6), User B pushes his state to the server again. When User C does the same, he first has to merge his state with the repository's state, resulting in state 7. Then User A pushes his changes (8), first having to merge with the repository's state as well, resulting in the comprehensive state 9. State 4 and 8 contain syntactic errors. Our Git server, which performs consistency checks on every submission, will not propagate these states as new ontology versions. The following algorithm shows how we extract the necessary information from the Git repository.

3.1. The Algorithm

Our algorithm is meant to create a provenance file for either the latest ontology file upload or a specific time interval. The following steps are taken for all ontology stages uploaded in the given time interval. In the first case, only the latest stage is processed.

3.1.1. Identifying Precursors for Comparison

To determine the exact changes made to RDF triples from one state to the next, we need to identify the state's precursors. Figure 1 shows all relevant cases: To get the changes made in state 2, we need to compare it to state 1. In order to get the changes in state 7, we have to compare it to state 5 as well as state 6. In case of an incorrect state, comparisons have to be made to their precursor. Thus, in order to get the changes in state 9, it is compared to state 2 and 7.

3.1.2. Extracting the Current and Precursors' State

To gather all content of the selected ontology states, we parse and translate all RDF data of one state into an EAV schema and save it in a delimited text file format.

3.1.3. Generating the Delta File

In order to compare two or more ontology states, we filter all text lines that occur in the new state but in neither of the precursors. This way, we collect all additions and, vice versa, we find removals by looking at what occurred in all precursors but not in the new state. Besides the delta files we also collect information about the ontology states (Git commit ID, Git parent commit IDs, timestamp, author's name, Git commit message) and the authors (author's name and author's e-mail-address).

3.1.4. Translation into Standardized Serialization

We decided to combine the highly interoperable PROV standard with the ChangeSet standard offering finer granularity. Both standards are published by the W3C [17,18]. The delta files, as well as the files containing authors and ontology states information, are translated into PROV-O files in turtle syntax.

The introduced algorithm extracts RDF statements that have been added or removed from one ontology version to the next. From this information, one may infer statements about what concepts have been added to, removed from or modified in the ontology. A concept's provenance information may be inferred without difficulty, as long as the RDF subject – in other words the concept's identifier – stays the same. If an identifier changed from one version to another, this information has to be stated explicitly in the RDF files.

4. Implementation

We perform the concept algorithm on a Debian Linux server with shell scripts and a java-based RDF parser: For extracting an ontology state, we parse its RDF files with Apache Jena Fuseki and save all available tuples through Fuseki's SPARQL interface

with the tool “curl”. Furthermore, we use standard tools like “grep” for delta file generation and “awk” to translate the delta files into provenance RDF.

4.1. Provenance File Content

Figure 2 shows example provenance statements we generated. The provenance information are partly extracted from the Git commit metadata via “git log”-command and partly from the delta files previously described. Git by itself offers the author username, e-mail-address, a 40-letter-ID for every commit, a timestamp and the previous commits’ IDs. The delta files contain the information whether a triple was added or removed, the affected concept (subject), attribute (predicate) and characteristic (object).

```
:ontology a prov:Entity .
:Stöhr a prov:Agent, foaf:Person ;
        foaf:mbox "mark.stoehr@innere.med.uni-giessen.de" .
:Majeed a prov:Agent, foaf:Person ;
        foaf:mbox "raphael.majeed@chiru.med.uni-giessen.de" .
:commit_6c365b51ef866b09f0a10e1e6b16546ffe746295 a prov:Activity ;
        prov:wasAssociatedWith :Stöhr ;
        prov:endedAtTime "2018-10-01T12:00:00+02:00"^^xsd:dateTime ;
        prov:label "Added concept for emphysema index." ;
        prov:actedOnBehalfOf :Majeed ;
        prov:wasInfluencedBy :commit_5af9416541b64f2471b54f5f40fe1d7cebafbe5f ;
        prov:generated [ a prov:Entity; prov:specializationOf :ontology ] ;
        prov:qualifiedUsage [
                a prov:Usage, cs:ChangeSet ;
                cs:addition [
                        a rdf:Statement;
                        rdf:subject :emphysemaIndex;
                        rdf:predicate rdf:type ;
                        rdf:object skos:Concept
                ] ;
        ] ;
        prov:used :emphysemaIndex .
:emphysemaIndex a prov:Entity .
```

Figure 2 – Example provenance statements in turtle N3 syntax. Git metadata in upper blue box, delta file data in lower green box.

Figure 2 illustrates most of the attributes we use in our provenance files. Additionally, we link two concept identifiers through the “prov:wasDerivedFrom” attribute with the new identifier as subject and the old identifier as literal value. Since this information cannot be extracted from the Git repository, it must be entered manually in the RDF files. Sometimes users edit the ontology on behalf of someone else. There is no dedicated field in Git that allows naming additional authors, but there are established conventions in the development community: the “commit message” is a field to describe the changes that were made. Co-authors may be appended to this message in a structured way.

4.2. Performance

In our DZL ontology’s first state at the end of 2016, it contained 637 statements. At the end of 2017, it contained 4069 statements. In November 2018, it contains 6081 statements. In total, 11866 additions and 5536 removals were performed. The overall size of our provenance files is approximated 4 megabytes. In its current state, one ontology version uses around 400 kilobytes of space.

The program for generating the provenance file from 2017-11-01 until 2018-11-01 takes 9 minutes and 19 seconds on our Debian Linux server. This includes loading into the triple store, exporting and comparing the versions, serializing and saving all extracted information. For the current single commit, this procedure takes 3.84 seconds.

4.3. Tracking Identifier Changes

When we implemented the algorithm, we retrospectively identified 67 concept identifier changes, which had to be annotated. We found them by searching through the Git repository's history tool, which lets users search for strings in added and removed lines. After implementation, changes in identifiers are annotated simultaneously during continuous ontology development. So far, 143 identifier changes have been annotated.

5. Lessons learned

In this article, we proposed a composition of standardized statements for provenance data representation. When looking at the W3 PROV documentation we find that a "prov:Activity" which modified a "prov:Entity" is usually meant to have "prov:used" an existing "prov:Entity" and "prov:generated" a new one. For our setup that would mean we had to keep a copy of every ontology version and its concepts. Since this information is stored in the repository, we decided to have only one "Entity" for every concept, which is referenced by the commit "Activity". Otherwise, we would create redundancy and file sizes would get out of hand.

In order to further simplify the curation process and make modifications even clearer, we consider classifying changes, e.g. structural, semantic and literal.

Our current work was focused on the algorithmic extraction, translation and standardized representation of provenance data. Its visualization is the next step to make information available for target users. There are several tools for visual representation of provenance. We will investigate in how far they can further support the curation process.

Since the program takes only few seconds for a single commit, it is reasonable to append it to Git's "after-push-hook" which already includes loading the data into our data warehouse. This way, all ontology data is always available together with complete provenance information.

Like other researchers, we found that blank nodes need a special treatment [11,8]. In our case, blank nodes would impede the comparison of ontology states. Thus, as a limitation, our algorithm demands that the underlying RDF data contains no blank nodes.

Concept identifier changes may lead to misinterpretation of the data delivered by our algorithm. Not only does the annotation of identifier changes need manual effort, but also new challenges may occur, e.g. if an identifier is reintroduced later for a different concept. Smart algorithms may identify derivations and reintroductions automatically with help of metric definitions and temporal relations.

In the introduction, we named several existing software solutions, of which we consider the Quit Store being the most advanced. Compared to other solutions, it provides detailed information about modifications down to the atomic RDF triple level, but to achieve this they introduced a proprietary namespace. Additionally, due to its

architecture, it is not applicable to a raw Git setup. All operations like merging and synchronization of data are done through the Quit API and the Repository Manager.

6. Conclusion

We proposed an enhanced composition of standard RDF statements for provenance data representation. Additionally, we developed an algorithm to extract and store provenance data from a Git repository accordingly. This algorithm works implementation-independent on top of ontology management system using Git and RDF. It can be applied to ontology versioning systems that are already in place, since the algorithm can backtrack changes up to the first ontology state. Because we use standard terminologies, the resulting data is reusable for any application with an appropriate standard interface. From the ontology-editing user's point of view, the development process remains the same except for additional steps in case of changes in identifiers. The software runs unnoticeably in the background. All source code is publicly accessible under Github: https://github.com/dzl-dm/cometar

Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] M.R. Stöhr, et al., CoMetaR: A Collaborative Metadata Repository for Biomedical Research Networks, *Studies in health technology and informatics* **245** (2017), 1337.
[2] M.R. Stöhr, R.W. Majeed, A. Günther, Using RDF and Git to Realize a Collaborative Metadata Repository, *Studies in health technology and informatics* **247** (2018), 556–560.
[3] M.D. Wilkinson, et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3** (2016), DOI: 10.1038/sdata.2016.18.
[4] P. Groth, et al., Requirements for Provenance on the Web, *International Journal of Digital Curation* **7(1)** (2012), 39–56.
[5] E.D. Ragan, et al., Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes, *IEEE transactions on visualization and computer graphics* **22(1)** (2016), 31–40.
[6] V. Curcin, et al., Implementing interoperable provenance in biomedical research, *Future Generation Computer Systems* **34** (2014), 1–16.
[7] B. Baum, et al., Opinion paper: Data provenance challenges in biomedical research, *it - Information Technology* **59(4)** (2017), DOI: 10.1515/itit-2016-0031.
[8] S. Auer, et al., A Versioning and Evolution Framework for RDF Knowledge Bases, *Perspectives of Systems Informatics* (2007), 55–69.
[9] L. Halilaj, et al., Git4Voc: Collaborative Vocabulary Development Based on Git, *International Journal of Semantic Computing* **10(02)** (2016), 167–191.
[10] T. de Nies, et al., Git2PROV: Exposing Version Control System Content as W3C PROV, *Poster and Demo Proceedings of the 12th International Semantic Web Conference* (1035) (2013), 125–128.
[11] N. Arndt, et al., Decentralized Collaborative Knowledge Management using Git, *Journal of Web Semantics* (2018), DOI: 10.1016/j.websem.2018.08.002.
[12] A.P. McGovern, et al., Glucose test provenance recording in UK primary care: was that fasted or random?, *Diabetic medicine : a journal of the British Diabetic Association* **34(1)** (2017), 93–98, DOI: 10.1111/dme.13067.

[13] S.S. Sahoo, J. Valdez, M. Rueschman, Scientific Reproducibility in Biomedical Research: Provenance Metadata Ontology for Semantic Annotation of Study Description, *AMIA ... Annual Symposium proceedings. AMIA Symposium* (2016), 1070–1079.
[14] R.S. Gonçalves, B. Parsia, U. Sattler, Ecco: A Hybrid Diff Tool for OWL 2 ontologies, *CEUR Workshop Proceedings* **849** (2012).
[15] E. Blomqvist, et al., EvoRDF: A Framework for Exploring Ontology Evolution, *The Semantic Web: ESWC 2017 Satellite Events* (2017), 104–108.
[16] Tim Berners-Lee, Dan Connolly, Delta: an ontology for the distribution of differences between RDF graphs, 2001 [Last accessed: 05/11/2018], https://www.w3.org/DesignIssues/Diff.
[17] W3C Semantic Web, DatasetDynamics/ChangeDescriptionVocabulary, 2011 [Last accessed: 05/11/2018], https://www.w3.org/wiki/DatasetDynamics/ChangeDescriptionVocabulary.
[18] W3C Semantic Web, PROV-O: The PROV Ontology [Last accessed: 05/11/2018], https://www.w3.org/TR/2013/REC-prov-o-20130430/.
[19] A.-K. Kock-Schoppenhauer, et al., Practical Extension of Provenance to Healthcare Data Based on the W3C PROV Standard, *Studies in health technology and informatics* **253** (2018), 28–32.
[20] P. Ciccarese, et al., PAV ontology: provenance, authoring and versioning, *Journal of biomedical semantics* **4(37)** (2013), DOI: 10.1186/2041-1480-4-37.

German Medical Data Sciences: Shaping Change – Creative Solutions for Innovative Medicine
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI190807

Verifying Data Integration Configurations for Semantical Correctness and Completeness

Mark R. STÖHR[a,1], Andreas GÜNTHER[a] and Raphael W. MAJEED[a]
[a] *UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany*

Abstract. Data integration is the problem of combining data residing at different sources and providing the user with a unified view of these data. In medical informatics, such a unified view enables retrospective analyses based on more facts and prospective recruitment of more patients than any single data collection by itself. The technical part of data integration is based on rules interpreted by software. These rules define how to perform the translation of source database schemata into the target database schema. Translation rules are formulated by data managers who usually do not have the knowledge about meaning and acquisition methods of the data they handle. The professionals (data providers) collecting the source data who have the respective knowledge again usually have no sufficient technical background. Since data providers are neither able to formulate the transformation rules themselves nor able to validate them, the whole process is fault-prone. Additionally, in continuous development and maintenance of (meta-) data repositories, data structures underlie changes, which may lead to outdated transformation rules. We did not find any technical solution, which enables data providers to formulate transformation rules themselves or which provides an understandable reflection of given rules. Our approach is to enable data providers understand the rules regarding their own data by presenting rules and available context visually. Context information is fetched from a metadata repository. In this paper, we propose a software tool that builds on existing data integration infrastructures. The tool provides a visually supported validation routine for data integration rules. In a first step towards its evaluation, we implement the tool into the DZL data integration process and verify the correct presentation of transformation rules.

Keywords. Quality improvement, metadata, automatic data processing, data accuracy, data aggregation, communication barriers, data visualization

1. Introduction

1.1. Background

Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data [1]. To harmonize multiple heterogeneous data sources is an important topic for research networks, registers and

[1] Corresponding Author, Mark R. Stöhr, UGMLC, Justus-Liebig-University, Klinikstraße 36, 35392 Gießen, Germany, E-Mail: mark.stoehr@innere.med.uni-giessen.de

other consortia planning data analyzes on large data pools. The complexity of data integration processes results not only from a technical point of view but also from the variety of professions whose contribution is required.

Future-oriented implementations of data integration processes have to meet various requirements in order to be able to interact with the global research community as well as to react to arising scientific questions. Wilkinson et al. formalized these requirements and introduced the FAIR principles (Findable, Accessible, Interoperable, Re-usable) as a guideline [2]. Expressiveness of data needs to be enriched with annotations, e.g. labels, standardized codes and hierarchical information, which guarantee unambiguousness and comparability. This can be achieved by establishing metadata repositories, which contain descriptive and relational information about the data repository [3–5].

It is not possible to set up a metadata catalogue once and expect it to meet all future requirements. Metadata is composed over time by domain experts and requirements to the matter addressed by data integration processes are growing with its use cases. Thus, an agile system development approach appears to be indispensable. For a single data collection, three technical components are involved in the data integration process: (1) the data source which is often historically grown and differs from other sources in structure, functionality and research scope. (2) The target data repository into which data from every source is to be integrated. (3) The metadata repository with descriptive information of the entries in the data repository.

While data source and metadata repository are not necessarily directly connected to each other, they are both connected to the data repository via an identifier. The technical operations to load source data into the data repository are executed by ETL (Extract, Transform, Load) software which extracts data from its source database, transforms it according to given rules and loads it into the target database.

To configure ETL software, knowledge of several different participants is needed. On the source side, there are *data owners*, *data providers* and *data managers*. Data providers collect and utilize data whereas data managers transform, store and export it. On the integration side, there are *coordinators* for the metadata repository development, *data managers* who transform and store the data and *software developers* providing and customizing the ETL software. While data managers come from a technical background, coordinators and users like physicians and medical documentalists have a medical background and the knowledge about the actual data acquiring processes.

Since every person involved is lacking knowledge from the other domains, mappings from source datasets to a harmonized dataset are fault-prone. Data providers are usually not able to compose formal transformation rules and data managers are often confronted with ambiguous data. After data managers composed the mapping rules, the respective data providers have to verify their correctness and completeness. This quality assurance is only possible, if the data provider understands the meaning of the technical realization. Besides semantic correctness, agile metadata development may cause changes affecting the ETL configuration. For example, a data element's temporary code may be replaced by a newly added code requested from a standardization organization. Therefore, data managers also have to check for newer versions of the metadata.

In this article, we address two issues impeding correct data integration: First, there is an indirect link between the data source and the metadata repository, which is established by the ETL configuration rules. This link is not transparent to data provider, since they have no access to the formal definitions. Thus, they cannot verify for

correctness and completeness. Second, data managers configuring the ETL mappings need a reflection of changes in the metadata that affect their configurations.

1.2. Requirements

To achieve solutions for the two identified issues, we need to provide transparency about the ETL process. We assume that an existing data integration infrastructure is already in place, including data sources, a metadata repository, a data warehouse and respective ETL processes. The same identifiers used in the data repository and to which source data is mapped are also available in the metadata repository. We also assume that changes to the metadata repository are available in processable form.

Every participant has to be able to retrieve the information they need in order to bring in their own expertise: Data providers must be able to verify the correctness and completeness of the composed mappings. Accordingly, they need to know how their data is mapped to the metadata catalogue. Thus, a tool is required to extract mapping information from the ETL configuration and visualize its binding to the metadata in an intuitively understandable way. All available context information from the metadata repository should be presented. This includes labels, descriptions, standardized codes and relations to other data elements like hierarchical information and "siblings". Additionally, the tool should indicate metadata changes, which affect the ETL configuration. The respective data managers must be able to understand the performed changes in order to update the configuration.

2. State of the art

We found several existing solutions targeting the integration of (clinical) data into data warehouses: Ong et al. [6] developed ETL software that uses ETL rule files that state the ETL mappings. They describe two different roles involved in the ETL process, so-called subject matter experts and database programmers. The subject matter experts are expected to "have extensive knowledge of the source and target schemas" and to be able to fill in rule entries. They admit, that for this process "certain SQL coding skill is needed" and "knowledge about the operators and functions of the DBMS is needed". The rules are entered in human-readable text-based files, since "GUI tools are not flexible enough and lack transparency". Pecoraro et al. [7] followed a similar approach by letting users enter rules in mapping files. They still require some technical background to perform this task. Additionally, their ETL software expects source data to be available in HL7 CDA and is only able to perform one-to-one and many-to-one mappings. Post et al. [8,9] developed a ETL software called Eureka!. They bypassed the need of varying mapping configurations by offering a stylesheet-based template in which source data has to be entered. The data-supplying user does not need to formulate relations of his data to the data repository. However, he is forced to put his data in a predetermined schema. With their setup, they realized "highly metadata-driven ETL processing". Vučković et al. [10] analyzed existing ETL modeling approaches and identified six basic transformation operations: Join, Equivalence, Equals, Concatenate, Condition and Constraint.

Additionally, there are generic tools for implementation and illustration of data integration processes, e.g. Talend Open Studio and Microsoft SQL Server Integration Services. They depict the transformation from source to target data schemas with various

operators like “merge” and “union”. However, they still lack sufficient clarity and context regarding the actual relations between data fields.

No presented solutions are applicable to scenarios in which physicians and medical staff are bound to their respective (clinical) input system and/or lack the required expertise to formalize mapping rules.

3. Concept

Our solution builds on top of existing infrastructure and complements the ETL configuration maintenance. Figure 1 is a schematic illustration of the intended infrastructure and workflow. In all ETL configurations, rules state how source fields are treated and how they will be represented in the target data schema. Those rules mostly follow the mindset of a database programmer and often contain cryptic terms and naming. Our approach is to move from the source data centered perspective to a target data centered one. That means, the required visualization tool displays the metadata repository’s elements with all their annotated information and for each of those elements, it illustrates their connection to the source data schema.

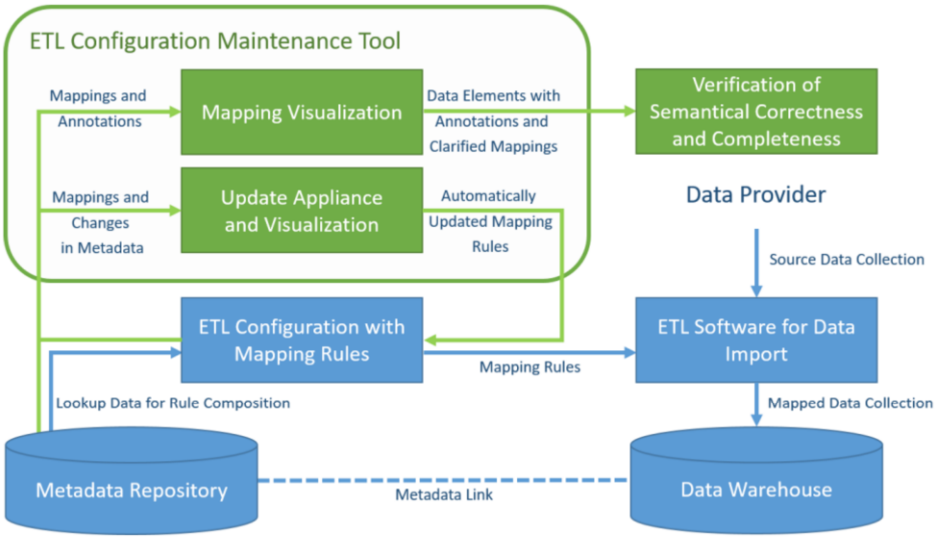


Figure 1. Intended infrastructure and workflow. Blue elements depict the ETL workflow for a single data collection. Green elements depict the additional verification process.

3.1. Available Information (Input)

There is no single solution for data integration infrastructures. In the introduction, we described three main components involved, the data source, the data repository and the metadata repository. Individually, they vary because of different purposes, histories, user capacities and preferences. To keep the concept as generic as possible, we assume that the metadata repository and the ETL configuration file provide the following abstract information in a processable way: The metadata repository stores annotations for each mapped source data field, e.g. labels, descriptions, standardized codes and

relations to other data elements. For each metadata element, we can access all changes that affect the ETL mappings, e.g. changes to the data element's datatype or even to its identifier. Furthermore, the composed ETL configurations include rules that can be categorized as shown in table 1 and table 2. These categories result from investigation (see state of the art section) as well as our own experiences based on over 20 integrated data sources.

Table 1. Categories of source to target data field transformations.

Field Transformation	Example
zero-to-one	Implicit information. Every patient in a cancer registry implies an existing cancer diagnosis, which is not explicitly mentioned.
one-to-one	Height of a person is a data element in both data schemas.
one-to-many	Diagnosis field filled with integers, each encoding the information for one diagnosis. A field containing information about both the biomaterial and its extraction type.
many-to-one	The data element "Operation and radiotherapy and chemotherapy" exists in the target data schema but each therapy is stored in a separate source field.
many-to-many	TNM classification in one field; the classification catalogue version in another field, which is required for a complete information. Both fields are mapped separately, but their semantic link remains.

Table 2. Categories of source to target data value transformations.

Value Transformation	Example
keep value as is	Height of a person is stored with unit centimeters in both data schemas.
calculate	Transform height in meters to height in centimeters by multiplication with "100".
lookup	If the source value is "1", then set value for concept "Pulmonary Hypertension" to true.
drop information	The concept/value is not part of the target data schema.

3.2. Presentation of Information (Output)

The data provider's view consists of metadata elements and mapping information. Metadata elements are shown with all available annotations. Mappings are all extracted from the ETL configuration file and presented as a combination of "Source Fields Names", "Source Value Condition", "Source Units" and "Target Value". Data managers also receive a list of changes made to the metadata, which affect their mappings. The tool offers them an automatically updated version of the ETL configuration.

4. Implementation

The German Center for Lung Research (DZL) uses i2b2 as its data warehouse software. Our ETL software expects source data to be available in delimited text format. The ETL configuration file is written in a proprietary XML format. The connection between metadata and the data repository is realized through concept codes, preferably standardized codes. For metadata composition, we use our Collaborative Metadata Repository (CoMetaR) tool. Metadata is expressed in RDF format and saved in a Git repository. Uploaded changes to the metadata are automatically propagated to the i2b2

ontology cell. Furthermore, metadata and its changes are available through a SPARQL interface. The CoMetaR browser (https://data.dzl.de/cometar/) visualizes the DZL ontology and lets users navigate and search through it.

We implemented the proposed ETL configuration maintenance tool in our CoMetaR browser. After uploading the configuration file through a file upload form, we show each mapping information under the respective metadata. Figure 2 shows an example where the field “Packyears” is mapped one-to-one to the DZL concept “Packyears” and non-empty values will be sent to the data warehouse without transformation. The field “Rauchverhalten” is mapped one-to-many to the DZL concepts “Active smoker”, “Ex-smoker” and others, depending on its string value. During the ETL process, the values are transformed through lookup from the respective string to “true”.

When uploading an ETL configuration file, all mapped concepts are automatically checked for outdated codes. If outdated, corresponding old and new codes are shown to the data manager and an updated version of the configuration file is offered for download.

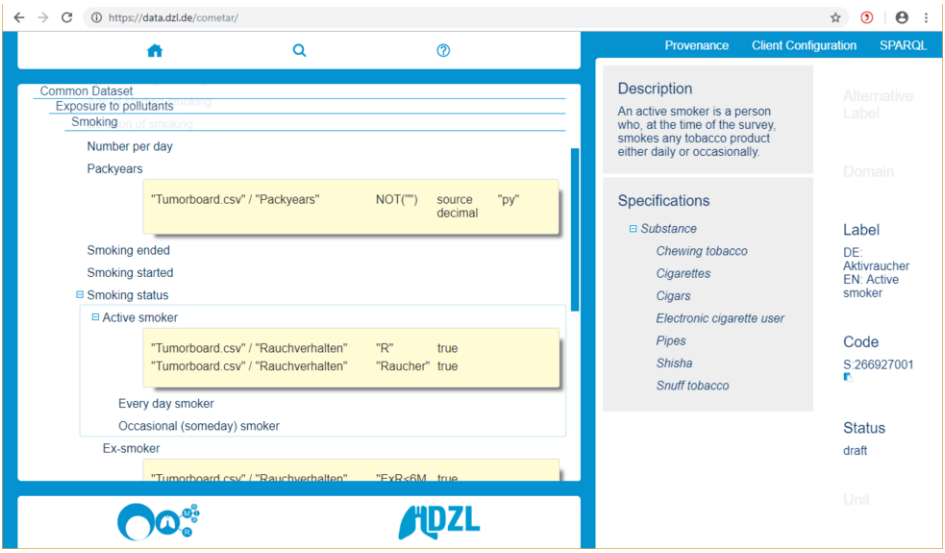


Figure 2. CoMetaR browser with loaded ETL configuration. To the left: hierarchically organized concept tree with ETL rules (yellow boxes). To the right: annotated metadata for the selected concept.

In order to achieve a comprehensive central data warehouse for the DZL, we already integrated more than 20 different data sources. This process includes the composition of mapping configuration files. In a first evaluation step for the proposed tool, we tested our implementation with the three largest of those configuration files containing 479, 256 and 158 mapping rules respectively. All rules are displayed correctly in the mapping visualization. However, our import software itself does not support many-to-one or many-to-many field transformations and no calculated value transformations.

5. Lessons learned (Discussion)

We introduced a new approach to analyze the quality of an ETL process regarding semantical correctness and completeness. Our claim is to enable data providers with less technical background to understand the formal definition of composed ETL mapping rules. Thereby, they could identify incorrect and/or missing mappings and inform the respective data managers. We also enable data managers to keep the ETL mappings up-to-date. We had to face some limitations when implementing our approach into our own infrastructure. Our ETL client can only use source files in delimited text format (e.g. CSV) and has limited support for many-to-one and many-to-many field transformations. Thus, for data integration we often need to pre-process data to meet our client's requirements. It is to check in how far mappings of pre-processed data are attributable to the original data source fields. In addition, an evaluation of the presented tool is needed in order to prove the concept's effectiveness.

6. Conclusion

We proposed the concept for a tool that supports data providers and data managers to keep ETL configurations for data integration processes correct, complete and up-to-date. Our approach is to display the metadata catalogue with all available annotations and highlight those, which are addressed by the ETL mapping rules. We provide transparency about how these complex rules operate on the source data. After verifying our implementation's ETL rule interpretation and presentation, we argue that additional context and rule clarification result in less ambiguity, more correct and complete mappings and thus higher data quality. We learned that preprocessing of source data might disguise the true origin of fields and values. Combined with the complexity of transformation rules, an intuitively understandable presentation of mappings can be challenging. Addressing this issue, we introduced a list of occurring mapping types and gave examples for a simplified visualization of their meaning regarding implicated source and metadata elements.

Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] M. Lenzerini, Data integration: a theoretical perspective (2002), DOI: 10.1145/543613.543644.
[2] M.D. Wilkinson, et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3** (2016), DOI: 10.1038/sdata.2016.18.
[3] H. Zhang, et al., An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival, *BMC medical informatics and decision making* **18(2)** (2018), 41, DOI: 10.1186/s12911-018-0636-4.
[4] V. Stathias, et al., Sustainable data and metadata management at the BD2K-LINCS Data Coordination and Integration Center, *Scientific data* **5** (2018), DOI: 10.1038/sdata.2018.117.
[5] M. Dugas, et al., ODMedit: uniform semantic annotation for data integration in medicine based on a public metadata repository, *BMC medical research methodology* **16** (2016), 65, DOI: 10.1186/s12874-016-0164-9.

[6] T.C. Ong, et al., Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading, *BMC medical informatics and decision making* **17(1)** (2017), 134, DOI: 10.1186/s12911-017-0532-3.
[7] F. Pecoraro, D. Luzi, F.L. Ricci, Designing ETL Tools to Feed a Data Warehouse Based on Electronic Healthcare Record Infrastructure, *Studies in health technology and informatics* **210** (2015), 929–933.
[8] A.R. Post, et al., Semantic ETL into i2b2 with Eureka!, *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science* **2013** (2013), 203–207.
[9] A.R. Post, et al., Metadata-driven Clinical Data Loading into i2b2 for Clinical and Translational Science Institutes, *AMIA Summits on Translational Science Proceedings* **2016** (2016), 184–193.
[10] M. Vučković, et al., The Specification of ETL Transformation Operations based on Weaving Models, *International Journal of Computers Communications & Control* **7(5)** (2012), 968–975, DOI: 10.15837/ijccc.2012.5.1356.

German Medical Data Sciences: Bringing Data to Life
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI210056

ISO 21526 Conform Metadata Editor for FAIR Unicode SKOS Thesauri

Mark R. STÖHR[a,1], Andreas GÜNTHER[a] and Raphael W. MAJEED[a]
[a] *UGMLC, German Center for Lung Research (DZL),*
Justus-Liebig-University, Giessen, Germany

Abstract. Metadata repositories are an indispensable component of data integration infrastructures and support semantic interoperability between knowledge organization systems. Standards for metadata representation like the ISO/IEC 11179 as well as the Resource Description Framework (RDF) and the Simple Knowledge Organization System (SKOS) by the World Wide Web Consortium were published to ensure metadata interoperability, maintainability and sustainability. The FAIR guidelines were composed to explicate those aspects in four principles divided in fifteen sub-principles. The ISO/IEC 21526 standard extends the 11179 standard for the domain of health care and mandates that SKOS be used for certain scenarios. In medical informatics, the composition of health care SKOS classification schemes is often managed by documentalists and data scientists. They use editors, which support them in producing comprehensive and valid metadata. Current metadata editors either do not properly support the SKOS resource annotations, require server applications or make use of additional databases for metadata storage. These characteristics are contrary to the application independency and versatility of raw Unicode SKOS files, e.g. the custom text arrangement, extensibility or copy & paste editing. We provide an application that adds navigation, auto completion and validity check capabilities on top of a regular Unicode text editor.

Keywords. Metadata, classification, data visualization, semantic web, data management, health information interoperability

1. Introduction

1.1. Background

Metadata repositories are an indispensable component of data integration infrastructures and support semantic interoperability between knowledge organization systems. The underlying controlled vocabulary varies in scope, notation format, storage type and complexity. This yields two key aspects for long-term functional appropriateness: First, the ability to interconnect with other metadata vocabularies in form of mappings and translations and secondly, the maintainability of the vocabulary itself. Common standards were published to approach these issues: The ISO/IEC 11179 standard defines a schema for representing metadata. It has been extended and clarified by the ISO/IEC 21526 standard to meet the requirements of healthcare [1,2]. The World Wide Web Consortium (W3C) defined multiple standards: The Resource Description Framework

[1] Corresponding Author, Mark R. Stöhr, UGMLC, Justus-Liebig-University, Klinikstraße 36, 35392 Gießen, Germany, E-Mail: mark.stoehr@innere.med.uni-giessen.de

(RDF) provides a common framework to share and reuse data across the semantic web [3]. The Web Ontology Language (OWL) provides a language to describe ontology resources, their relations and properties expressed in RDF [4]. The Simple Knowledge Organization System (SKOS) is a lightweight data model formally defined as OWL ontology with purpose to describe thesauri and other classification schemes [5]. It is considered a compromise between poorly structured taxonomies and extensively formalized ontologies. A large collection of example SKOS datasets can be found on the W3C homepage, showing that SKOS has a wide scope of application including national libraries, social sciences, MeSH terms and drug administration forms [6]. The ISO/IEC 21526 standard explicitly "mandates the use of SKOS to provide user-interface surfaced content classification". The use of SKOS leads directly to metadata with improved compliance with three of the four FAIR principles [7]: Resources are registered with unique identifiers (Findability), data is available in a "broadly applicable language for knowledge representation" (Interoperability) and resources are annotated with preferred labels, descriptions and notations (Reusable). The fourth principle "Accessibility" can be achieved by loading SKOS resource definition files into a dedicated application, e.g. a triple store like Apache Jena Fuseki with SPARQL interface [8,9].

Implementing the SKOS standard for representing a classification scheme means to use one or more of RDF's serializations formats, the most popular ones being RDF/XML, Notation3, Turtle and JSON. We prefer Turtle, which is highly human readable and still machine parsable at reasonable costs. Since it is written as Unicode text, it can on the one hand easily be shared between people and applications and on the other hand be maintained in simple text editors. Nevertheless, especially when handling large classification schemes, the latter can lead to negative user experiences: Resource definitions and their relations become confusing, syntax errors will possibly be unrecognized until the documents are parsed by a machine. All known tools for maintaining SKOS data make use of (server) applications with additional databases and user interfaces to operate on them. Although the tools offer interfaces that allow the import and export of various Unicode formats, they thereby negate the simplicity, independency (no server needed, offline editing possible) and high flexibility (e.g. extensibility, copy & paste editing, commenting) of raw Unicode SKOS data. The decision for an editing tool depends on the target users. Conway et al. claim that in the field of medicine such tools are mostly used by physicians [10], whereas we made contrary experiences: In the domain of medical informatics, the development of classification schemes necessitates a certain amount of knowledge about metadata management and compliance with design patterns. For example, hierarchical subordinated elements may inherit context information, different catalogues in different versions need to be merged (e.g. TNM classification) and annotated codes may require accurate post-coordination. Therefore, in many cases editing is performed by data managers and medical documentalists. For such clientele, an editor should support users at their task while not impeding any Unicode text editor functionality.

In this paper, we elaborate a solution to fill a disregarded niche of Unicode SKOS classification scheme maintenance.

1.2. Requirements

Throughout over one hundred iterations of editing SKOS data and uploading it into our metadata repository, we identified various requirements for an assisting Unicode editor: (1) We need RDF syntax highlighting as well as syntax verification, (2) a presentation

of the defined SKOS hierarchy and (3) the possibility to include multiple Unicode files at once. The editor should (4) support navigation within the classification scheme: selecting a node in the concept tree or selecting a concept's reference to another concept should navigate to the respective text segment. Hovering a concept identifier should (5) offer context information about the SKOS resource, e.g. label and hierarchical classification. (6) Changes should be adopted immediately and (7) proper suggestions (auto completion) should be offered during typing. (8) We require basic semantic verification, e.g. checks for unique preferred labels, an existing English label and no loops in hierarchy. Improper text segments should be visualized for the user. For the purpose of sustainability and due to limited resources, we do not want to develop a completely new editor from scratch. Thus, we require (9) a platform independent existing text editor with extension capabilities. The editor itself should work (10) server-independently/stand-alone, which on the one hand corresponds to the proven approach of editing offline and publishing/versioning online and on the other hand prevents accessibility issues like server downtime or network problems.

2. State of the art

Although we seek a server-independent editor, we still want to take the most prominent ones and their SKOS editing capabilities into account: Protégé, developed by the Stanford University, incorporates a large toolset for complex OWL ontology maintenance [11]. Like its browser-based counterpart "Web Protégé" [12], Protégé only considers the "subclass" and "is a" relation for hierarchy evaluation. Thus, SKOS' "broader" and "narrower" relations are supported, but they will not result in a proper tree view. An SKOS plugin was developed in 2011, but it is not supported by the latest Protégé version and development has been discontinued [13]. The University of Utah developed a web-based SKOS editor, which unfortunately is no longer accessible [10]. Additionally, there are various commercial vendors like PoolParty [14] and TopQuadrant. Those focus on creating and maintaining thesauri according to the SKOS standard. They offer additional tools like text mining, graph visualization and use deep learning algorithms for content classification. Again, the actual data is either stored on a web server or in a binary database.

On the other hand, we have a large amount of text editors. To name only few popular ones, there are Atom, Notepad++, Vim and Sublime. Many of them provide an extension interface, making them potential candidates for powerful SKOS editors.

3. Concept

The most appealing text editor is the Visual Studio Code editor by Microsoft. It is free to use, has a strong community, is highly extensible, platform independent, provides built-in Git commands, the IntelliSense feature offers smart completion capabilities and our working group already made positive experiences with it. Additionally, an extension for RDF syntax highlighting and syntax verification is already available. Visual Studio Code extensions are written in typescript.

In order to build a Visual Studio Code extension that meets the requirements, we seek to make use of as many built-in features as possible. Defining SKOS classification schemes has lot in common with program code writing: Both have clear grammar rules

and include references to other text segments. Visual Studio Code offers interfaces to implement custom features for code editing, context information, validation and navigation, e.g. “Go to Reference”, search and replace, “Tree View” panels, tooltip on hover, word suggestion, diagnostics and word completion (“IntelliSense”).

We will make use of many of SKOS’ classes (“Concept”, “Collection”, “ConceptScheme”), hierarchical relations (“broader”, “narrower”, “inScheme”, “member”, “topConceptOf”, “hasTopConcept”) and properties (“prefLabel”, “notation”).

Regarding the overall architecture of our extension, the user interface consists of two panels: The actual text editor and the tree view, which depicts the SKOS resources in hierarchical order. Editing the text document will trigger the extension to parse the whole text document and update these definitions and the tree view. We will implement a button to load all turtle files from the current working directory. During text editing, the user will be offered suggestions based on the context, e.g. after typing “skos:broader” the extension will suggest all known concepts. Hovering a resource will show helpful information in a tooltip, e.g. the label and the hierarchical classification. Using the “Right Click → Go to Implementation” function as well as clicking a resource in the tree view will jump to its defining paragraph(s). Using the “Right Click → Go to Reference” function will show all text segments that reference the respective resource. We use the Visual Studio Code “Diagnostic Collection” to highlight invalid text segments with proper severity level.

For text document parsing, we decided to use regular expressions. We defined them according to the RDF grammar rules (https://www.w3.org/TR/turtle/#sec-grammar-grammar). The downside of this approach is that regular expressions do not support recursion. To solve this issue, we decided not to parse nested blank nodes. Throughout our research, we did not come across any thesaurus making use of nested blank nodes.

4. Implementation

During development, two users intensively tested the editor for usability, performance and functional appropriateness. As data basis, we made use of various publicly available Turtle SKOS metadata sets.

Figure 1 shows some of the extension’s key features: On the left side is a tree view panel, which shows all defined “Concept”, “Collection” and “ConceptScheme” instances in a hierarchical arrangement. Clicking on a tree view item will focus the respective text segment or offer a selection if more than one segments describe the item. Located on the right side is an outline panel. The middle section shows the actual text editor with active syntax highlighting. Green text sections are comments that will be ignored by the parser and can be used to take notes or to blank out definitions. After typing “skos:broader”, the editor offers a list of all available preferred labels and additionally a descriptive window showing the selected item’s hierarchical classification and all defining text paragraphs. Picking a selection will insert the selected item’s identifier.

To prove the editor’s functionality, we loaded three different SKOS datasets: (1) The “STW Thesaurus for Economics” by the Leibniz Information Centre for Economics [15], (2) the UNESCO Thesaurus [16] and (3) the German Center for Lung Research (DZL) thesaurus [17]. The first two thesauri can be downloaded and both consist of one Unicode file with approximately 3.4 megabyte and 3.2 megabyte in size. The DZL thesaurus consists of 63 Unicode files with a total of approximately 700 kilobyte. All

thesauri were parsed successfully and displayed correctly. To test the loading performance, we used a computer with an i7-8700 CPU. Loading the STW thesaurus took approximately 59 seconds, the UNESCO thesaurus approximately 20 seconds and the DZL thesaurus took less than 2 seconds. For comparison: Loading the STW thesaurus into an Apache Jena Fuseki server instance took less than one second. After every text editing, the refreshing takes the same amount of time. During that refresh process, interaction with the tree view is delayed, but text editing including suggestions still works seamlessly. Additionally, we wanted to test the extension for the "NIH NLM Value Sets" thesaurus found in BioPortal [18], but had to realize that with over 37 megabytes in size it was too large for our parser in its current version.

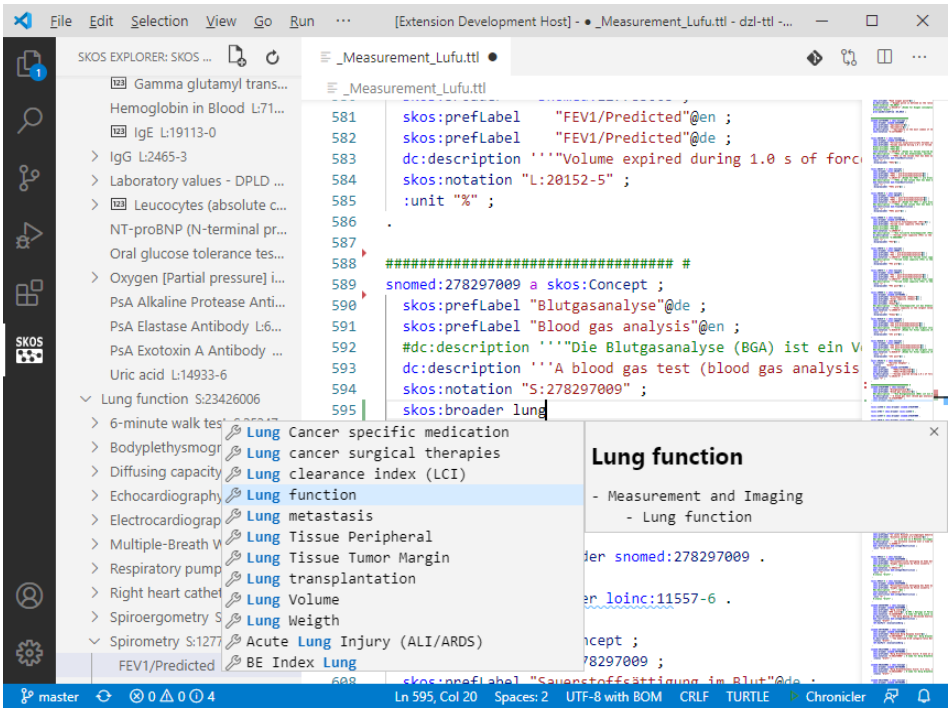


Figure 1. Screenshot of the editor's user interface. Left side: Tree view. Middle: Text editor. During text input, a suggestion window with additional context information pops up.

The implementation in its current version has been tested for usability by a medical documentalist and a medical information scientist. Both are experienced in editing text-based SKOS thesauri. They became a short introduction into the developed editor's features. For testing, the DZL thesaurus has been extended by a disease area specific catalog. The users shared their experiences in a short qualitative evaluation interview naming the advantages and disadvantages they found. Beneath the ability to navigate through the thesaurus, the users especially appreciated the semantic checks. Proper indication for duplicates and ambiguous preferred labels raised awareness for concepts that are included in more than one sub-catalog. Auto completion for known SKOS resources helped preventing typing errors. On the other hand, navigating with the "Go to Reference" and "Go to Implementation" features turned out to be less intuitive for non-software architects.

5. Lessons learned (Discussion)

The Visual Studio Code documentation helped us to develop an editor extension that meets our requirements entirely for relatively small classification schemes. For larger thesauri (up to a size of around 3 megabytes or 7000 SKOS resources) like the "STW Thesaurus for Economics" and the UNESCO Thesaurus, the functionality is assured, but maintenance is not as fluent. This is because the editor is not yet built for performance. After every change, the whole file is parsed again. Thus, refreshing is delayed and editing feels less fluent. For smaller thesauri, the user experience is positive, especially the semantic verifications have been found very useful. An additional benefit that occurred during tests was the fact, that the built-in Git module automatically indicated changes made since the last thesaurus upload. Of course, this only works, if the thesaurus is versioned with Git. When searching for datasets to test our application, we found many thesauri in RDF XML format. For more compatibility, additional parsers should be developed. The current parser only works for Turtle files and ignores nested blank nodes. During our research, we did not find any thesaurus making use of nested blank nodes. Another suggestion for future development could be an additional form-based interface for editing SKOS resources. Visual Studio Code provides so-called "Web Views" for custom HTML-based interfaces.

6. Conclusion

We identified the necessity to complement the collection of SKOS thesaurus editors with a server-independent standalone Unicode SKOS thesaurus editor. Our main purpose was to keep the versatility of raw Turtle formatted text files (e.g. custom text arrangement, extensibility, copy & paste editing, versioning) and to add navigation, auto completion and validity check capabilities on top. For thesauri divided in multiple files, the presented editor works as intended. It offers benefits for usability like clarity, validity checks and fast navigation. For larger thesauri, improvements in performance are required. The extension is publicly accessible:
https://marketplace.visualstudio.com/items?itemName=markstoehr.skos-ttl-editor

Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] S.M.N. Ngouongo, M. Löbe, J. Stausberg, The ISO/IEC 11179 norm for metadata registries: does it cover healthcare standards in empirical research?, *Journal of biomedical informatics* **46(2)** (2013), 318–327, DOI: 10.1016/j.jbi.2012.11.008.

[2] International Organization for Standardization, ISO/TS 21526:2019 Health informatics — Metadata repository requirements (MetaRep) [Last accessed: 25/02/2020], https://www.iso.org/standard/71041.html.

[3] World Wide Web Consortium, Resource Description Framework, 2014 [Last accessed: 25/02/2020], https://www.w3.org/RDF/.

[4] World Wide Web Consortium, Web Ontology Language, 2012 [Last accessed: 25/02/2020], https://www.w3.org/OWL/.

[5] World Wide Web Consortium, Simple Knowledge Organization System, 2009 [Last accessed: 25/02/2020], https://www.w3.org/2004/02/skos/.

[6] World Wide Web Consortium, SKOS/Datasets, 2018 [Last accessed: 26/03/2020], https://www.w3.org/2001/sw/wiki/SKOS/Datasets.

[7] M.D. Wilkinson, et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3** (2016), DOI: 10.1038/sdata.2016.18.

[8] The Apache Software Foundation, Jena Fuseki, 2011 [Last accessed: 14/07/2020], https://jena.apache.org/documentation/fuseki2/.

[9] W3C Semantic Web, SPARQL Protocol And RDF Query Language (SPARQL), 2013 [Last accessed: 14/07/2020], https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[10] M. Conway, et al., Developing a web-based SKOS editor, *Journal of biomedical semantics* **7** (2016), 5, DOI: 10.1186/s13326-015-0043-z.

[11] T. Tudorache, et al., Supporting Collaborative Ontology Development in Protégé, *ISWC 2008: The Semantic Web* (2008), 17–32.

[12] M. Horridge, et al., WebProtege: a collaborative Web-based platform for editing biomedical ontologies, *Bioinformatics (Oxford, England)* **30(16)** (2014), 2384–2385, DOI: 10.1093/bioinformatics/btu256.

[13] S. Jupp, S. Bechhofer, R. Stevens, A Flexible API and Editor for SKOS, *ESWC 2009: The Semantic Web: Research and Applications* **2009**, 506–520.

[14] T. Schandl, A. Blumauer, PoolParty: SKOS Thesaurus Management Utilizing Linked Data, *ESWC 2010: The Semantic Web: Research and Applications* (2010), 421–425.

[15] Leibniz Information Centre for Economics, STW Thesaurus for Economics, 2019 [Last accessed: 25/02/2020], http://zbw.eu/stw/version/latest/about.

[16] UNESCO, UNESCO Thesaurus [Last accessed: 25/02/2020], https://skos.um.es/unescothes/.

[17] German Center for Lung Research, CoMetaR - Collaborative Metadata Repository, 2019 [Last accessed: 25/02/2020].

[18] M. Salvadores, NIH NLM Value Set as a SKOS terminology, 2015 [Last accessed: 27/03/2020], https://bioportal.bioontology.org/ontologies/NLMVS.

Original Paper

The Collaborative Metadata Repository (CoMetaR) Web App: Quantitative and Qualitative Usability Evaluation

Mark R Stöhr; Andreas Günther, Prof Dr; Raphael W Majeed

Justus-Liebig-University Giessen, Universities of Giessen and Marburg Lung Center (UGMLC), German Center for Lung Research (DZL), Gießen, Germany

Corresponding Author:
Mark R Stöhr
Justus-Liebig-University Giessen
Universities of Giessen and Marburg Lung Center (UGMLC)
German Center for Lung Research (DZL)
Klinikstraße 36
Gießen, 35392
Germany
Phone: 49 641 985 42117
Email: mark.stoehr@innere.med.uni-giessen.de

Abstract

Background: In the field of medicine and medical informatics, the importance of comprehensive metadata has long been recognized, and the composition of metadata has become its own field of profession and research. To ensure sustainable and meaningful metadata are maintained, standards and guidelines such as the FAIR (Findability, Accessibility, Interoperability, Reusability) principles have been published. The compilation and maintenance of metadata is performed by field experts supported by metadata management apps. The usability of these apps, for example, in terms of ease of use, efficiency, and error tolerance, crucially determines their benefit to those interested in the data.

Objective: This study aims to provide a metadata management app with high usability that assists scientists in compiling and using rich metadata. We aim to evaluate our recently developed interactive web app for our collaborative metadata repository (CoMetaR). This study reflects how real users perceive the app by assessing usability scores and explicit usability issues.

Methods: We evaluated the CoMetaR web app by measuring the usability of 3 modules: *core module*, *provenance module*, and *data integration module*. We defined 10 tasks in which users must acquire information specific to their user role. The participants were asked to complete the tasks in a live web meeting. We used the System Usability Scale questionnaire to measure the usability of the app. For qualitative analysis, we applied a modified think aloud method with the following thematic analysis and categorization into the ISO 9241-110 usability categories.

Results: A total of 12 individuals participated in the study. We found that over 97% (85/88) of all the tasks were completed successfully. We measured usability scores of 81, 81, and 72 for the 3 evaluated modules. The qualitative analysis resulted in 24 issues with the app.

Conclusions: A usability score of 81 implies very good usability for the 2 modules, whereas a usability score of 72 still indicates acceptable usability for the third module. We identified 24 issues that serve as starting points for further development. Our method proved to be effective and efficient in terms of effort and outcome. It can be adapted to evaluate apps within the medical informatics field and potentially beyond.

(JMIR Med Inform 2021;9(11):e30308) doi: 10.2196/30308

KEYWORDS

usability; metadata; data visualization; semantic web; data management; data warehousing; communication barriers; quality improvement; biological ontologies; data curation

Introduction

The Importance of Metadata

Raw data are useless without metadata that characterizes and contextualizes its content. A number is meaningless without the information on which parameter it describes (eg, blood pressure) and a finding is of no use without its context (eg, sepsis as a comorbidity vs sepsis as cause of death). Metadata itself always needs context (eg, the concept it describes). In many cases, metadata are merely implied by column headers of tabular databases and the implicit knowledge of the few people working with the database. Many information scientists have researched the field of metadata, for example, Wilkinson et al [1], who published the FAIR (Findability, Accessibility, Interoperability, Reusability) principles, which is a guideline for well-designed metadata. Whenever data are reused (for analysis, validity checks, etc), the corresponding metadata must be attached to the actual data. Thus, explicitly formulated, rich, and comprehensive metadata are indispensable for any sustainable research project [2]. At present, most data processing is done automatically by computers, which necessitates all metadata to be available in machine-readable form [3]. In addition to data processing, metadata are used to describe data sets to a broader audience, such as the national or international research community. BioPortal [4], for example, is a comprehensive repository of biomedical ontologies interconnecting researchers globally. In addition, there are approaches for recording the variety of existing data and metadata repositories in public registers [5,6].

Metadata in the Field of Data Integration

Overview

Particularly in the context of data integration within large research networks, comprehensive metadata are essential. “Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data” [7]. Although the process of exporting, transforming, and loading data is a huge task, this *unified view* is an achievement by itself. In medical informatics, the purpose of data integration is to promote translational research and to have access to a larger data pool for retrospective data analysis and prospective patient recruitment. The amount of integrated data and the way they are presented to users determine their acceptance and accessibility. If too few concepts are covered by a repository or if too few instances of data are integrated, researchers have no sufficient basis for analysis. If metadata are not presented accessibly, users will presume app shortcomings rather than investing in exploration time. This applies especially to entry-level users and, in most cases, yields in rejection of the software.

Data Integration: Main Components and Roles

Software-driven data integration involves multiple technical components: various *heterogeneous source databases* are harmonized and integrated into a *collective data repository*. All affected parameters, more precisely the canonical concepts behind these parameters, are annotated in a separate *metadata repository*. Both repositories are linked through identifiers [8-10]. *Configuration files* define the harmonization process of different source database schemata into a target schema. These configuration files vary in format and syntax, but all of them are written in a formal computer-readable language [11-14].

From the user perspective, these components are managed and elaborated by the following roles: *data providers* know the meaning of their data and its acquisition processes. In medical informatics, this knowledge is essential for data harmonization, because labels such as column names or form labels are not always sufficiently specific. According to Nadkarni and Marenco [15], “[...] column names may be quasi-gibberish, heavily abbreviated, and their names may follow arbitrary conventions that are idiosyncratic to the system designer or organization.” Rahm and Bernstein [16] showed that even automatic schema matching can only provide mapping candidates. The formulation of mapping rules is performed by the *local and central data managers* (responsible for the source databases and collective database repository, respectively) as they have the required technical background to maintain the formally written configuration files. *Data coordinators* elaborate the metadata repository content, incorporating multiple studies and registers with varying scopes and the focus of research. This process includes rating for relevance, harmonization, annotation, curation, and clustering. The clustering and hierarchical organization of metadata have a direct impact on the presentation of user interfaces. It determines how intuitively information can be found and used.

Information Access Barriers

To provide a data warehouse with comprehensive and accurate data, different roles need access to different classes of information residing in the described data integration system. We identified 3 cases in which access barriers prevent users from contributing their expertise [17,18]:

1. All users need access to the listing of all data elements represented in the data warehouse. These annotations and context information can be derived from the metadata repository and must be visualized.
2. Data managers and, in particular, data providers need full access to the mapping rules for data harmonization. They are only available in the formal language, which requires the respective information technology background. Data providers usually do not have that knowledge.
3. Data coordinators need access to the provenance information of the metadata to be able to curate it. “Especially in collaborative metadata development, a comprehensive annotation about ‘who contributed what, when and why’ is essential” [17].

In most cases, barrier (1) is resolved through metadata browsers [4,19,20]. For metadata repositories in the context of data integration, barriers (2) and (3) often form a huge gap between users and the required information.

The Implementation of Collaborative Metadata Repository

The German Center for Lung Research (German: Deutsches Zentrum für Lungenforschung [DZL]) implemented the collaborative metadata repository (CoMetaR), applying

principles of collaborative metadata development and FAIR metadata warehousing [1,17,18,21]. It is based on open and commonly used standards. The DZL metadata constitutes a highly specified thesaurus specifically developed for lung research, and till July 2021, it contains 3.474 distinct concepts. CoMetaR supports storing a single thesaurus in the Resource Description Framework (RDF) format based on the Simple Knowledge Organization System (SKOS) and Dublin Core (DC) knowledge organization systems [22-24]. The ISO/IEC 21526 [25] standard explicitly "mandates the use of SKOS to provide user-interface surfaced content classification." Versioning occurs via Git, which also provides information about the changes among different versions [26,27]. The latest thesaurus version is loaded in a triple store and accessible through the SPARQL Protocol and RDF Query Language (SPARQL) interface [28]. This interface can be used to extract metadata information and, as in our case, to set up tree-like metadata in a data warehouse [29]. The extracted metadata information can also be used to generate a visual metadata representation similar to our user front end, the CoMetaR web app. This front end was developed to dissolve access barriers for all user roles and thereby support them in contributing to their expertise. However, it has yet to be proven scientifically that the CoMetaR web app meets the requirements for metadata management and data integration support.

This study evaluates the usability of 3 modules built for common tasks in the field of data integration and metadata maintenance.

Methods

Study Design

Overview

The usability evaluation performed was a combination of (1) the think aloud method and (2) usability questionnaires. By combining both methods, we wanted to measure both observable and perceived usability. The execution consisted of two phases: (1) a screen sharing–supported training specific to the respective user's roles and (2) solving of the given tasks by the participant with subsequent retrospection, including the completion of a usability questionnaire. All evaluations were performed by the same experimenter.

The Think Aloud Method

This method is commonly applied to the usability evaluations of web interfaces [30,31]. The idea behind the think aloud method is that participants verbalize their thoughts while performing given tasks. Their expressions were recorded and later transcribed and analyzed according to an interpretation model.

We decided not to record the participants but to make notes on their expressions as well as their app use behavior. These notes focused on usability, functional, and methodological issues. The advantage of this approach is a more comfortable setting for the user on the one hand and less effort for the experimenter on the other hand. The downside is the potential information loss because the experimenter already filters information.

As our interpretation model, we used the 7 categories described in ISO 9241-110 [32]: suitability for the task, conformity with user expectations, suitability for learning, suitability for individualization, self-descriptiveness, controllability, and error tolerance.

System Usability Scale

We used the System Usability Scale invented by Brooke in 1996 as a measurement tool for the usability of the app. This scale was introduced as a *quick and dirty* but a meaningful measurement tool for user experience [33,34]. It consists of 10 questions answered on a scale from 0 to 4. All questions are available in multiple languages, including German, which we used for our evaluation.

Materials

CoMetaR Modules

The CoMetaR web app is divided into a concept tree navigation area and a module area. Modules can be selected in the module menu in the top-right corner, as shown in Figure 1. In the following paragraphs, we will briefly describe the functionality of the 3 evaluated modules: the *core module*, *provenance module*, and *data integration module*. In the *Introduction* section, we described 3 user roles involved in the data integration process: data managers, data providers, and data coordinators. A user may perform more than one role. Each role makes use of the core module, whereas the data integration module and provenance module are more role-specific (see the Tasks section).

The core module functionality of the CoMetaR web app (Figure 1) involves browsing through all metadata concepts and showing the corresponding detailed information. Users can navigate the concept tree by expanding the nodes and retrieving details by clicking them. They can also use the search function to check if and where a concept is located in the thesaurus. Concept details are shown in the module area. They include core information like labels, alternative labels, data type, code, status (*on draft* yes or no), and unit. In addition, we present the author, description, and concept specifications. A dedicated panel shows the history of all changes that have been made to the selected concept. A button allows the export of the concept and all of its subconcepts with basic information in the CSV format.

As our metadata are growing and developing over time with many participants involved, we decided to provide the provenance module, which enables users to track all changes. These changes may be the additions, moves, or removal of concepts in the concept tree, but also modifications of their annotations. When selecting the provenance module (Figure 2), the affected concept tree elements receive icons that symbolize their changes for a given timespan. The default timespan is 1 month from the current date and can be adjusted in the module. The module itself shows all dates concerned with metadata changes in vertical order. Horizontal bars attached to such a date represent single uploads, and their width indicates the amount of change. Clicking a date or a single upload bar loads the respective changes and shows them in the concept tree underneath the corresponding concepts.

Figure 1. Screenshot of the collaborative metadata repository (CoMetaR) web app core module. Left side: concept tree. Right side: module content (concept details). Top-right corner: module navigation. Top-left corner: home button, search panel, and help panel.

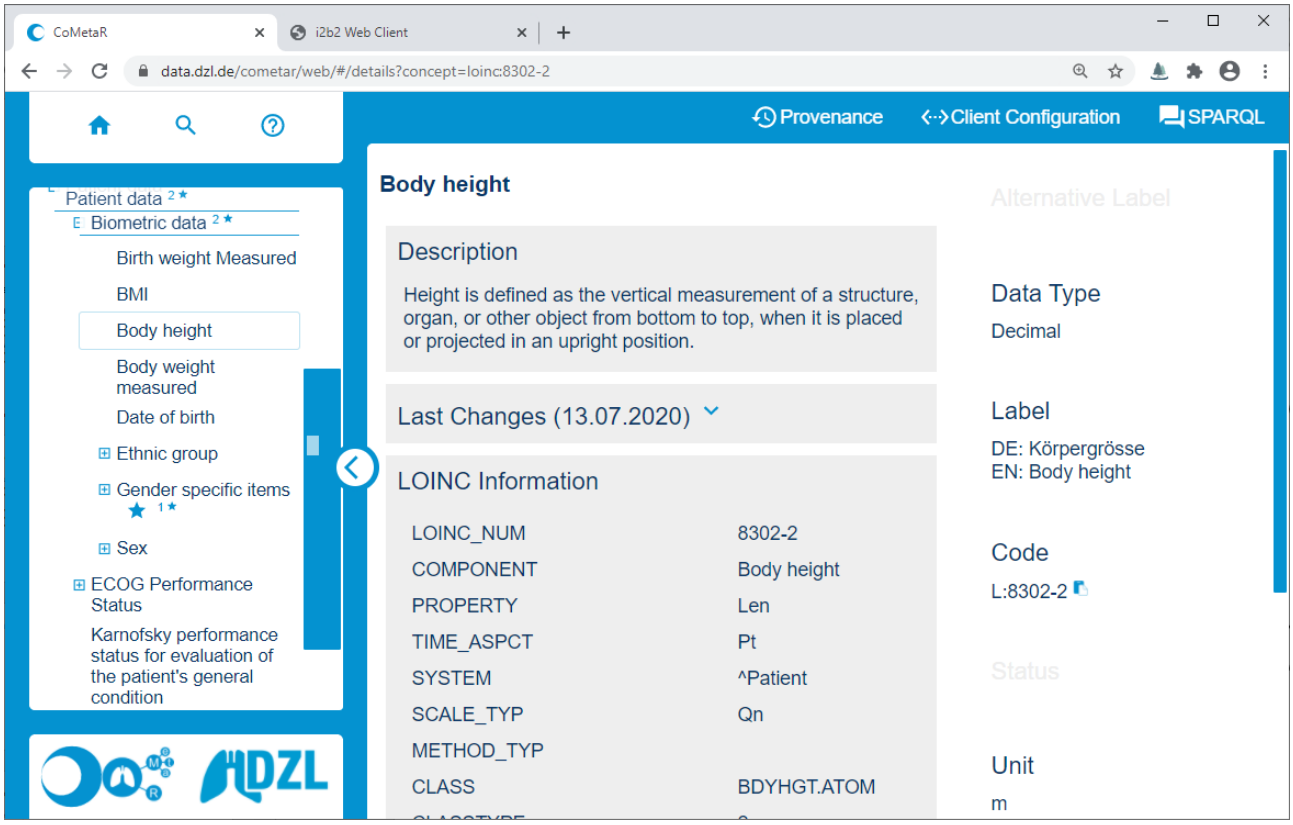
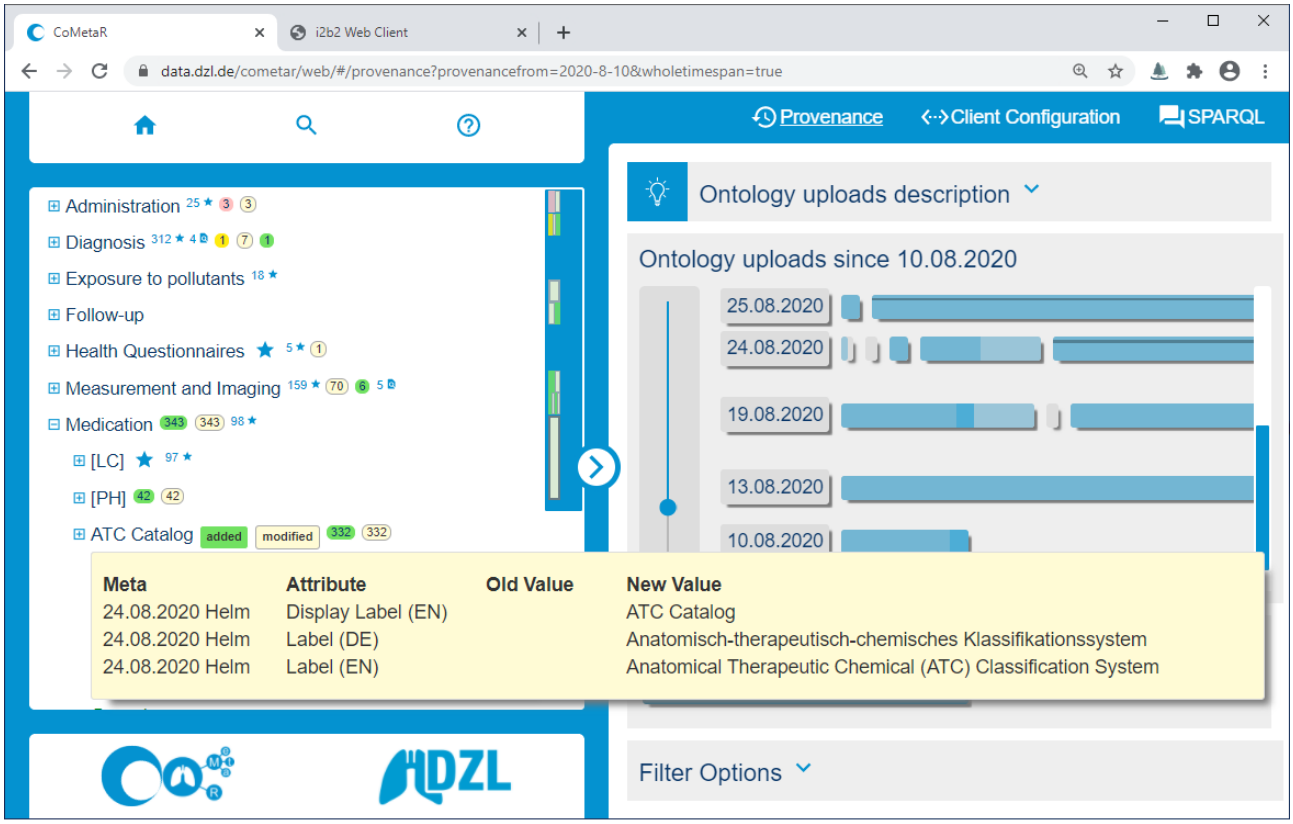
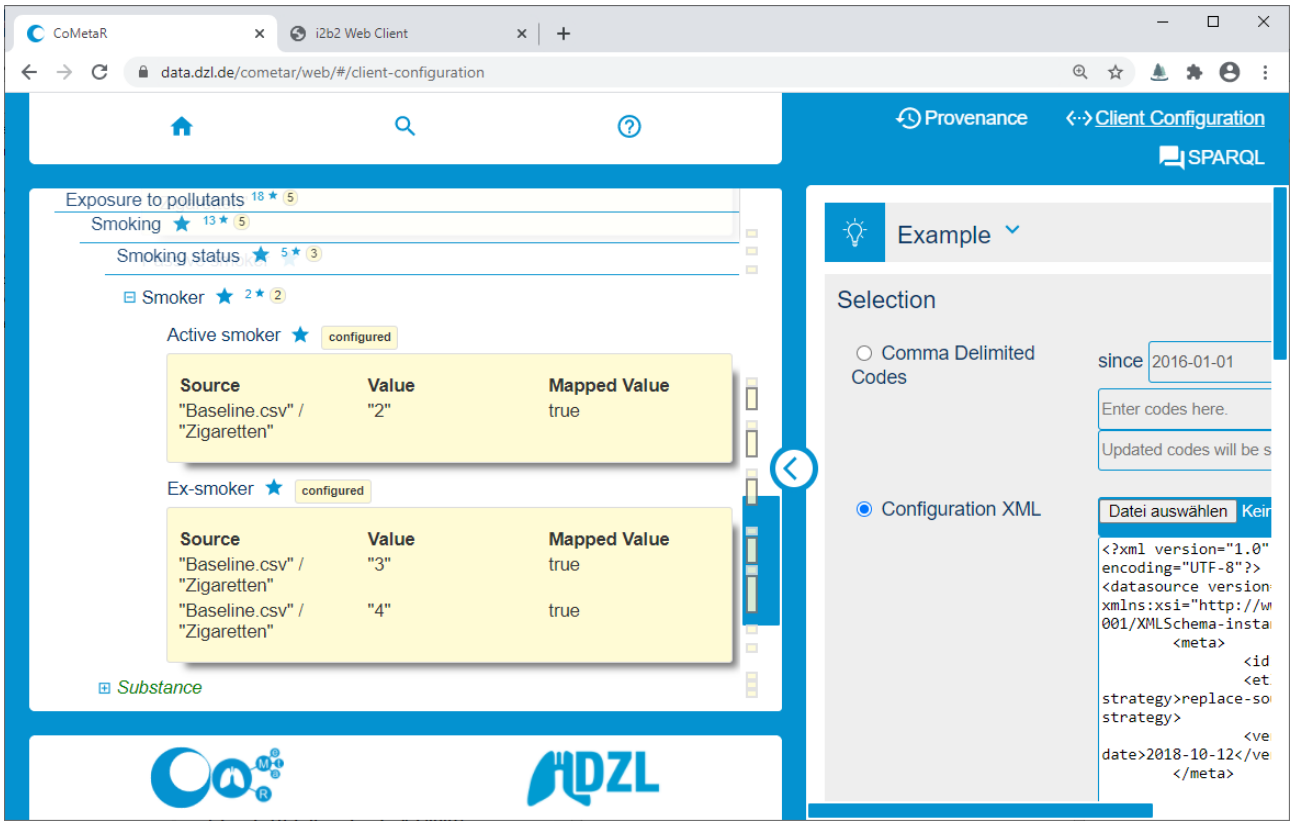


Figure 2. Screenshot of the collaborative metadata repository (CoMetaR) web app provenance module. Left side: concept tree with colorized annotations for added, moved, or removed and modified items. Light yellow box: information box for the item ATC Catalog on mouse-over. Right side: module content (upload history visualization). ATC: Anatomical Therapeutic Chemical.



Our data integration process is supported by the data integration module. The integration process for a single data source is divided into 4 parts. (1) The export of data from the source system, (2) the preparation of data for the integration software, (3) configuration of the integration software, and (4) its execution. As the configuration file is written in formal language to be interpreted by software, it is not accessible for humans who lack the required technical background. To verify the configurations, the respective data providers must be able to access the formulated rules. For this task, they can upload the configuration file to the data integration module (Figure 3). All rules are then shown below the corresponding concept in the concept tree. In addition, we print notifications in the module area if any rule refers to a concept that does not exist (anymore) or that has been reintroduced. In such a case, the correct reference can be determined automatically, depending on the metadata's formal documentation. Subsequently, an updated version of the configuration file is offered for download. Note that this process does not invoke any kind of data upload; it is solely used to verify the configuration itself.

Figure 3. Screenshot of the collaborative metadata repository (CoMetaR) web app data integration module. Left side: concept tree. Light yellow boxes: corresponding mapping rules. Right side: module content (configuration file upload).



Tasks

CoMetaR was designed to support data integration tasks. In the German Center for Lung Research, we have been practicing data integration since 2016 and identified information that is of high interest for data integration experts. For example, to match and map elements of the source data to the integrated data, the person formulating the rules needs to know which elements are part of the integrated metadata, what are their exact characteristics (method of measurement, scale, classification, etc), and how they are uniquely identified. If these characteristics change, the mapping rules must be adjusted. For various processes, people often want the metadata to be available in Microsoft Excel format, yielding the need for respective export capabilities. For these and further scenarios, we defined 10 tasks that verified CoMetaR's suitability in the field of lung research. The following tasks were composed by 2 experts, who have been internationally active in the field of data integration for >5 years. The composition process included brainstorming, discussion, and finally consensus. To assign modules to each participant, we considered their user roles as well as their everyday tasks. All users must solve core module tasks, all data coordinators must solve provenance module tasks, and all data managers who upload data must solve the data integration module tasks.

The first 4 tasks aim at the use of the core module. They test the ability to search for and find specific thesaurus elements and their annotations as well as the capability to export data:

1. Indicate which of the parameters *Never smoker* and *Opportunity smoker* are part of the DZL metadata.

2. Indicate code, datatype, and unit of the spirometry parameter Forced Expiratory Volume in 1 Second (*FEV1*) according to the metadata.

3. Regarding the last change of the concept *Comorbidities*, indicate its date and the modifications applied.

4. Describe in detail which individual steps you would take to print the subtree of *Biometric Data* in tabular form.

The following 2 tasks aim at the use of the provenance module. They test the ability to track changes within the thesaurus:

5. Indicate which concepts have been added, moved, or removed in the last month.

6. Pick one concept for which annotations have been changed in the last upload. Indicate who performed this change on which date.

The last 4 tasks aim at the use of the data integration module. They test the ability to verify individual upload client configurations:

7. Examine the configuration for falsely mapped concepts.

8. Examine the configuration for properly mapped concepts.

9. Examine the metadata for concepts that are not mapped in the configuration but you could provide.

10. Update your local configuration to meet changed concept references. Describe your approach.

Tasks 7, 8, and 9 must be seen as one task with 3 subtasks. The participants were asked to use their own configuration files designed for uploading the data they administered. Some configuration files comprise hundreds of mapping rules. Depending on the size and coverage of certain data sources, task fulfillment takes a considerable amount of time. During the live evaluation, the participants were asked to work on each of these 3 tasks exemplarily to be able to fill out the System Usability Scale questionnaire. They completed the tasks asynchronously and reported their results when they finished.

Configuration Files

For 3 of the 4 data integration module tasks, we asked the participants to use their own configuration file for analysis. These comprise rules to define how local concepts are mapped to concepts in the central data warehouse. The file format is XML. The configuration files are used by a data transformation and upload client software. Configuration files do not contain any instance data. By using real configuration files instead of an artificial example, we were able to test our app in a realistic scenario and identify faulty mappings. In addition, this setup allowed participants to work with familiar information.

Experimenter Notes

The experimenter completed a notes sheet alongside following the evaluation procedure. It was structured to contain one row per participant and the following columns: *Experience level*, *English level*, *age*, *profession*, *roles* (see the *Introduction* section), *evaluation date*, *training start timestamp*, *training finished timestamp*, *notes for training*. Each of the 3 modules contains the following columns: *module tasks* (stating whether tasks were solved successfully), *module finished timestamp*, *notes for module*, *timestamp module questionnaire filled*.

System Usability Scale Questionnaires

The questionnaires handed to the participants contained 10 usability questions defined in the System Usability Scale. They were put into a Microsoft Excel sheet with one row for each question and columns for values of 0 to 4. The final score for the 10 questions was calculated within the sheet. The participants were handed one sheet per evaluated module.

Quantitative Analysis Sheet

A spreadsheet was used to collect the scores per participant and module to calculate the quantitative analysis parameters, that is, *range from*, *range to*, *mean score*, and *SD*. These 4 parameters were additionally calculated with respect to the participant's experience level, using the following formula:

$$\text{Score weighted by experience} = \text{score} - 4 \times (\text{experience level} - 1)$$

Given an experience level from 1 to 5, the score weighted by experience differs by up to 16 points, which corresponds to previous findings [35]. In addition to scores from the questionnaires, the corresponding experience levels, and the calculated values, no participant-related information was put into the sheet.

Setting

To evaluate our web app, we decided to interact with the participants remotely (participants were not invited to a local test laboratory) and synchronously (the evaluator and participant executed the test session in real time). We made one exception for a very time-consuming task type, which certain participants completed asynchronously. This method appeared to be the most efficient in terms of preparation effort, travel time, and risk of SARS-CoV-2 infection. Its suitability was shown in a comprehensive study: Bastien [36] summarized multiple studies stating that remote evaluations yield comparable results with a local laboratory evaluation. Although he found that automatic recording of every user interaction with the app can provide more insights about the app's usability, the setup is very time-consuming and would only be rational for larger participant numbers. The participants were approached in April and May of 2020. Data collection took place in May and June of 2020. Data analysis was conducted in July of 2020.

As a communication platform, we used the GoToMeeting web conference software by LogMeIn [37]. It allows participants to dial in via phone or software app. The latter also offers screen sharing capabilities, which all but one participant with technical issues were able to use.

Sampling

The target audience of CoMetaR is experts who contribute to the task of data integration as data providers, data managers, or data coordinators. Our implementation of CoMetaR is dedicated to lung research. Therefore, in this evaluation, we included members of the German Center for Lung Research and collaborating organizations. The included participants should cover a wide range of roles and responsibilities. These characteristics determine the module that they can work on effectively. For example, data managers who load data into a data warehouse have a data integration configuration file and can use the data integration module. The core module is relevant to all the user roles. In contrast, the provenance module is mostly relevant for data coordinators and data managers, whereas the data integration module is mostly relevant for data managers and data providers. In addition to their user role, profession,

age, and English level, we also asked for the participants' experience with the app. English and experience levels were measured on a scale of 1 to 5.

Bastien [36] cited studies showing that most usability problems can be found in 5-15 participants. As Virzi [38] showed, only 4 to 5 participants were needed to identify about 80% of all usability issues, and this number is enough to reveal the most severe issues. Therefore, we planned to recruit at least 5 participants for each aspect of the web app. In total, we approached 13 potential participants, of whom 12 agreed to participate.

Ethical Considerations

All methods were performed in accordance with the relevant guidelines and regulations. This study was granted an exemption from requiring ethics approval by the ethics committee of the Faculty of Medicine at the Justus-Liebig-University in Giessen, Germany. Informed consent for participation in the study was obtained from all the participants.

All patient-related data were recorded anonymized. It covers age, profession, role, evaluated modules, English level, and experience with the app. The data were further coarsened using age classes of 10 years to prevent participant reidentification.

Procedure and Data Collection

Overview

Before any evaluation, we performed a screen sharing–supported training specific to the respective user's roles, regardless of previous experiences with the app. The goal of this training was to provide participants with equal basic knowledge about the web app's structure and functionality. We asked for the participants' previous experiences with the system, which may influence the evaluation outcome [35]. After the training, the participants shared their screens and completed the tasks given by the evaluator. After using the module, the participants filled out the System Usability Scale questionnaire. This also gave them the chance for retrospection and a short dialogue with the experimenter, potentially revealing more usability issues.

Instructions

After giving each participant introductory training regarding the app's functionalities, they had the option to ask questions and clarify misunderstandings. Following, for each tested module, they were asked to fulfill each task one by one. The tasks were communicated via speech. The experimenter asked the participants to verbalize their thoughts during the evaluation and reminded them whenever they forgot. After the participant solved the tasks for a module, the experimenter asked them to fill out the usability questionnaire we sent them previously via email. Furthermore, they were invited to participate in a retrospective dialogue, again noting the findings.

Role of the Experimenter

The experimenter played a passive role. During the evaluation, he was not supposed to speak besides reminding the participant to verbalize their thoughts. In cases where the participants were stuck, the experimenter gave hints to lead to the information that had to be received from the app. Meanwhile, the experimenter completed the structured notes sheet documenting the participants' verbalized thoughts, spontaneous reactions, and their app use behavior, focusing on the previously mentioned usability categories [32].

Recording and Transcription

The traditional think aloud method requires recording the entire evaluation session and the following transcription. As mentioned in the study design, we did not record sessions because transcription occurred during the session.

Analysis

Quantitative Analysis

For quantitative analysis, we calculated aggregated scores (*range from*, *range to*, *mean score*, and *SD*) based on the System Usability Scale questionnaires. We additionally calculated the same aggregations factoring in the experience level. This adjustment is motivated by previous findings, which show that usability scores vary up to 16 points based on the participant's experience level [35]. For example, a user with no experience (level 1) has the same base and adjusted score, whereas a user with a score of 70 and experience level 4 (of 5) has an adjusted score of 58. By calculating these moderated scores, we hope to obtain better insights into the app's usability, especially regarding entry barriers. All calculations were performed using Excel (see *Materials* section). We omitted subgroup analysis by English level, age, and profession as our sample size was too small.

Qualitative Analysis

We conducted a thematic analysis of the information gathered during the evaluations to identify usability issue patterns and to present a descriptive account of users' experiences. After familiarization with all notes, we went through all notes again and generated usability issue statements. We followed a latent approach, which means that we interpreted the data to create statements that were more meaningful. For example, task 2 asked the participants to indicate the properties of the spirometry parameter *FEV1*. In one case, a participant used the search function and entered *Spiro FEV1*, which led to no results (a note in the experimenter's structured notes file). Our conclusion is not that our app is unable to find a specific pattern but that users expect a more powerful search functionality, as is known from bigger internet companies (theme). After generating usability issue themes, we combined similar statements and reviewed them by checking if all notes were still well-represented by these statements. These were then assigned to 1 of the 7 usability categories described in ISO 9241-110 [32]: suitability for the task, conformity with user expectations, suitability for learning, suitability for individualization, self-descriptiveness, controllability, and error tolerance. The categorization was performed by the same person who underwent the evaluation sessions with all participants. Afterward, these groupings were discussed internally with another expert and potentially adjusted.

Software

For documentation and analysis, we used only Microsoft Excel and Microsoft Word.

Quality Assurance

The System Usability Scale questionnaire consists of 10 questions, 5 of which stated a positive usability and 5 of them stated negative usability. As some questions include negations, we assumed a possible misinterpretation. Therefore, we immediately checked each questionnaire for outliers and inquired when we identified potential misinterpretations. When inquiring, we again pointed out that we do not insist on better scores but on valid answers.

We wanted to ensure correct and comprehensive categorization, as well as unambiguous wording for qualitative analysis. A second person who was familiar with the study design and aspects of usability checked all categorizations. The resulting tables are the results of in-depth dialogues.

Results

Participants

All participants in this evaluation currently work for or in collaboration with the German Center for Lung Research. Their operation areas and responsibilities vary, but all contribute to the data integration task. Table 1 shows the details of all the 12 participants. They vary in age (28-63 years), experience with the system (1-4 on a scale of 1-5), English level (2-5 on a scale of 1-5), and profession (medical documentalists, medical informatics specialists, graduated biologists, bioinformatics specialists, study coordinators, and data managers).

Table 1. Characteristics of the 12 participants including age, experience level, English level, profession, user roles, and tested modules.

Characteristics	Participants											
	A	B	C	D	E	F	G	H	I	J	K	L
Age (years)	30-40	30-40	30-40	40-50	50-60	60-70	30-40	50-60	30-40	50-60	60-70	20-30
Experience level (1-5)	3	3	4	2	4	3	3	3	3	1	2	4
English level (1-5)	3	3	4	3	4	4	4	5	3	3	2	4
Profession	MD[a]	DM[b]	MI[c]	SC[d]	MD	GB[e]	MI	DM	DM	MD	MD	BI[f]
Has role data manager	✓[g]	✓	✓	✓				✓	✓	✓	✓	✓
Has role data provider	✓	✓	✓			✓						
Has role data coordinator	✓	✓	✓		✓	✓	✓					
Tested core module	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tested provenance module	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓
Tested data integration module	✓	✓	✓	✓					✓			

[a]MD: medical documentalist.

[b]DM: data manager.

[c]MI: medical informatics specialist.

[d]SC: study coordinator.

[e]GB: graduate biologist.

[f]BI: bioinformatics specialist.

[g]Characteristic present.

Quantitative Analysis

Time Expenditure

The training took between 10 and 30 minutes, depending on how many modules were presented and how many questions the participants had. After training, for task completion, the core module took between 8 and 26 (average 14, SD 6) minutes. The provenance module took between 3 and 20 (average 9, SD 5) minutes. The configuration module took between 21 and 51 (average 37, SD 12) minutes. Regarding the latter, we did not include the time spent asynchronously to complete the tasks.

Usability

Each participant solved the tasks of one or more CoMetaR modules (core module n=12, provenance module n=10, data integration module n=5). Subsequently, they completed one System Usability Scale questionnaire separately for each module. According to Bangor et al [39], a mean score of 50.9 or higher can be seen as *OK*, a mean score of 71.4 or higher is *Good*, and a mean score of 85.5 or higher is *Excellent*; a mean score of 70 or higher indicates that the interface is acceptable. The System Usability Scale score results are presented in Table 2. For *weighted by experience*, we subtracted up to 16 points based on the user's own perceived experience.

Table 2. Aggregated System Usability Scale scores.

Module and score type	Values, mean (SD; range)
Core module	
Usability score	81.5 (9.1; 60.0-92.5)
Weighted by experience	73.8 (7.8; 60.0-84.5)
Provenance module	
Usability score	72.3 (16.0; 37.5-90.0)
Weighted by experience	63.9 (15.20; 37.5-79.5)
Data integration module	
Usability score	81.0 (9.9; 65.0-92.5)
Weighted by experience	73.0 (9.9; 57.0-84.5)

Functional Suitability

All the participants successfully solved all given tasks. In total, 12 participants solved 48 core module tasks, 10 participants solved 20 provenance module tasks, and 5 participants solved 20 data integration module tasks. In the case of task 2, 2 participants did not find the correct tree node and needed a hint. During the provenance module tasks, 1 participant lost track because he loaded too much information from multiple modules into the tree. He needed a hint to reset the app to solve task 5. In total, 97% (85/88) of all tasks were solved independently.

Qualitative Analysis

Our thematic analysis led to 24 usability issue themes, which covered all functional inadequacies and complications identified during the experiment. We grouped these themes into the 7 categories described in ISO 9241-110 (Textboxes 1-5). As the app does not offer possibilities for individualization, the respective category *suitability for individualization* does not appear in this evaluation. None of the observed issues were assigned to the category *controllability*.

Textbox 1. Issues in the category Suitability for the Task.

Core module

- Using the search function for *FEV1* shows more than 100 results because it is used as criterion for many diseases. Most of the results are located in the comorbidities-subtree.
- The help window does not help with task 2.

Provenance module

- The mouse-over tooltip of upload bars sometimes distracts and overlays other bars.
- Changing the selection of upload bars leads to changes in the concept tree. The system gives insufficient feedback that these changes were applied.

Textbox 2. Issues in the category Conformity with User Expectations.

Core module

- The search function only searches for fixed substrings and does not behave comparably to a mighty World Wide Web search engine. This might lead to incorrect conclusions whether a concept is part of the metadata.
- The users expected the fixed headings for the currently displayed subtree to be interactive.

Provenance module

- The provenance module disappears when clicking a tree element and the element's core information are shown instead.

Textbox 3. Issues in the category Suitability for Learning.

Core module

- An element's change history is part of the core module and not the provenance module.
- Structural information for elements (added, moved, or removed) are not explicitly displayed in the element's history (last changes).
- The number of search matches is not the number of matched concepts but of all matched attributes.
- Some annotations like *added* have rectangular representation in the minimap or outline and round-cornered representation in the tree.

Provenance module

- The structural annotations (added, moved, or removed) refer to the selected provenance timespan and not only to the selected uploads.
- It is not intuitive that a moved element's old and new concept tree position are both selected when clicking one of them.

Textbox 4. Issues in the category Self-Descriptiveness.

Core module

- Many people search the code for *Forced Expiratory Volume in 1 Second* in the *Logical Observation Identifiers Names and Codes (LOINC)*–description instead of the concept's core information.
- For some users, it is not intuitively clear that details for a tree node are shown when clicking them.
- Symbols in the tree are not explained through a legend, but only mouse-over tooltips.
- The minimap or outline next to the scrollbar is not intuitive for users that are not familiar with such.
- The scroll bar is differently styled than a standard scroll bar and might not instantly be recognized as such.

Provenance module

- For some users, it is not noticeable whether an upload was selected.
- The function of the *load all changes* button is not clear.
- The temporal order (left to right or right to left) of multiple uploads on the same day is not clear.

Data integration module

- For elements with more than one configuration rule, it is not intuitive that the rules are applied from top to bottom order.

Textbox 5. Issues in the category Error Tolerance.

Core module, provenance module, and data integration module

- Activating multiple modules and searches leads to an overload of information in the concept tree.
- Loading too many information into the tree and expanding many of affected tree elements leads to high central processing unit (CPU) use.

Discussion

Principal Findings

In total, 12 participants took part in the evaluation of up to 3 modules of the CoMetaR web app, and each participant completed up to 10 tasks; 97% (85/88) of all tasks were solved independently and successfully. The core module and data integration module both obtained a mean usability score of 81, which proves good and nearly excellent usability. For inexperienced users, we estimated a mean usability score of 73, which proves good and acceptable usability. The provenance module has a mean usability score of approximately 72, which implies good and acceptable usability. For inexperienced provenance module users, we estimated a mean usability score of 63, which indicates unacceptable usability. We identified 24 issues with the app, which we grouped into 5 usability categories based on ISO 9241-110. From our point of view, of particular note are (1) information displayed in the concept tree can be overwhelming, especially if information from multiple modules is shown at once. (2) For many users, the provenance module and its functionalities are not accessible. The number of options, such as filtering by timespan or upload package, demand an extensive introduction and learning period. (3) The search functionality can output far more hits than expected because every literal information about concepts is considered. Some sort of categorization or filtering may be useful.

Strengths and Limitations

The strength of our study design is the relationship between effort and outcome. Although we omitted the step of recording audio and video of each session, we found a considerable compilation of usability issues and clear quantitative categorization of our tested modules owing to the System

Usability Scale questionnaire. All testing sessions were performed by a single experimenter. For thematic analysis, an additional scientist was consulted.

Retrospectively, we identified 4 problems regarding the evaluation methodology. The web conference software used in this evaluation was always visible and, in some cases, overlapped crucial information in the browser window. Second, one person tried to participate via an Apple product and was not able to establish screen sharing because of missing technical literacy. The third problem concerns communicational logistics, specifically around task instructions being communicated verbally by the evaluator. Some participants missed important aspects of the tasks because they were inattentive or started solving the tasks before the instruction was finished. Finally, some tasks were not formulated in sufficient detail. For example, for task 5, a participant thought it would be sufficient to read the respective upload description, but we expected them to list all changes explicitly in detail.

We did not record audio and video, for which reason we probably missed single verbalizations and observations. Thus, we cannot claim that our list of usability issues is complete at 100%, which arguably is never the case. In addition, the experimenter already filtered information during the test sessions, which might have biased the qualitative analysis outcome. We still assume that we found most usability issues, especially the most severe ones, because the experimenter was able to follow every action throughout all sessions without difficulty.

As all tasks were performed in our production environment, the upload history and thus the collection of added, moved, or removed or modified concepts varied. This may have led to differing results among the participants. We assumed that these differences were negligible in the usability evaluation.

Comparison With Previous Work

In 2009, considering 317 web apps, Bangor et al [39] found that web apps have a mean usability score of 68.2, which confirms the above-average usability of our app. Owing to increased awareness regarding usability, these values might have changed, but we did not find a more recent usability score meta-analysis. To the best of our knowledge, our approach to calculate another score for inexperienced users has not been done before. It allows the assessment of usability scores for inexperienced or new users even though some participants already have experience with the app.

Regarding the think aloud method, it is usual to record and transcribe all user sessions. Other studies show that this consumes a considerable amount of time and labor, which is often done by multiple scientists. In addition, we did not count code quantities within a transcript, as this is often done in a thematic analysis. We adopted the highest-level themes from an ISO standard instead of creating them ourselves.

Implications and Future Work

After evaluating our app, we are able to improve it by addressing all found usability issues. This will, in the first place, improve research in the field of lung research because lung research–specific metadata availability and accessibility will be improved. This app has already been considered by other German Centers for Health Research. We hope to be able to generally improve the field of health research.

Second, we applied a methodology that allows the usability evaluation of metadata management apps with a considerably low effort in time and labor. In an adapted form, this method can be applied to similar apps. Although the first 4 tasks of our evaluation are specific to the field of lung research concerning content, their content-agnostic intention is to check if basic information can be retrieved from the app. This includes the existence and findability of concepts (task 1), identification of a concept's annotations (task 2), its development over time (task 3), and the export of information about a unit of concepts (task 4). The application programming interface for the data integration module is specific to our data integration configuration file format, but the tasks represent the crucial steps to be taken to verify such a configuration file. The next step for this project could be the application of this evaluation method to comparable apps to approve its reliability and to find common usability issues.

We also hope that the findings of our qualitative analysis raise other developers' awareness of possible shortcomings in their own apps. For example, they might also plan to visually annotate concepts in the concept tree, in which case we highly recommend not displaying too much information at once.

A potential alternative or addition to the think aloud method with a thematic approach could be a heuristic evaluation performed by usability experts. The advantages and disadvantages of both methods were researched by Yen and Bakken [30].

We experienced issues with the web conference software, whose control panel sometimes overlapped crucial information on the user display. For further remotely and synchronously performed evaluations, we recommend ensuring that all relevant web app content is always visible, for example, by choosing different conference software.

We found that the assumed average usability score for inexperienced users was approximately 8 points lower than the original average score. This implies, on the one hand, that entry barriers exist within the app. On the other hand, these barriers can at least partly be overcome with experience. Measuring such a score might be of special interest for apps that provide a more efficient alternative to existing methods of information retrieval. Entry barriers may lead to rapid rejection of the entire software.

Conclusions

Our goal was to find usability issues of the CoMetaR web app and to measure its usability as perceived by real users. We identified 24 issues, which will be starting points for app improvement. On average, the app was assessed as good and in parts nearly excellent in terms of usability. Our method proved effective and efficient in terms of effort and outcome. Future research should improve our app and evaluate similar solutions. We invite other researchers interested in evaluating biomedical metadata repositories to adapt our methodology.

All source codes are publicly accessible under GitHub [40]. Research metadata repository is publicly accessible [41]. The production instance of the German Center for Lung

Acknowledgments

The German Center for Lung Research (German: Deutsches Zentrum für Lungenforschung) is funded by the German Federal Ministry of Education and Research (German: Bundesministerium für Bildung und Forschung). Marc Griffiths proofread the paper as a native English speaker.

All data generated or analyzed during this study are included in this published paper.

Authors' Contributions

MRS developed the collaborative metadata repository software, which was evaluated in this study. MRS and RWM elaborated on the study design, including the composition of tasks. MRS performed the interviews with all participants and interpreted the data. RWM and AG substantively revised the study during all steps.

Conflicts of Interest

None declared.

References

1. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 2016 Mar 15;3:160018 [FREE Full text] [doi: 10.1038/sdata.2016.18] [Medline: 26978244]
2. Kush RD, Warzel D, Kush MA, Sherman A, Navarro EA, Fitzmartin R, et al. FAIR data sharing: the roles of common data elements and harmonization. J Biomed Inform 2020 Jul;107:103421 [FREE Full text] [doi: 10.1016/j.jbi.2020.103421] [Medline: 32407878]
3. Hume S, Chow A, Evans J, Malfait F, Chason J, Wold JD, et al. CDISC SHARE, a global, cloud-based resource of machine-readable CDISC standards for clinical and translational research. AMIA Jt Summits Transl Sci Proc 2018 May 18;2017:94-103 [FREE Full text] [Medline: 29888049]
4. Noy NF, Shah NH, Whetzel PL, Dai B, Dorf M, Griffith N, et al. BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res 2009 Jul;37(Web Server issue):W170-W173 [FREE Full text] [doi: 10.1093/nar/gkp440] [Medline: 19483092]
5. Sansone S, McQuilton P, Rocca-Serra P, Gonzalez-Beltran A, Izzo M, Lister AL, FAIRsharing Community. FAIRsharing as a community approach to standards, repositories and policies. Nat Biotechnol 2019 Apr;37(4):358-367 [FREE Full text] [doi: 10.1038/s41587-019-0080-8] [Medline: 30940948]
6. Pampel H, Vierkant P, Scholze F, Bertelmann R, Kindling M, Klump J, et al. Making research data repositories visible: the re3data.org Registry. PLoS One 2013 Nov 4;8(11):e78080 [FREE Full text] [doi: 10.1371/journal.pone.0078080] [Medline: 24223762]
7. Lenzerini M. Data integration: a theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2002 Presented at: SIGMOD/PODS02: International Conference on Management of Data and Symposium on Principles Database and Systems; Jun 3 - 5, 2002; Madison Wisconsin. [doi: 10.1145/543613.543644]
8. Zhang H, Guo Y, Li Q, George TJ, Shenkman E, Modave F, et al. An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. BMC Med Inform Decis Mak 2018 Jul 23;18(Suppl 2):41. [doi: 10.1186/s12911-018-0636-4] [Medline: 30066664]
9. Stathias V, Koleti A, Vidović D, Cooper DJ, Jagodnik KM, Terryn R, et al. Sustainable data and metadata management at the BD2K-LINCS Data Coordination and Integration Center. Sci Data 2018 Jun 19;5:180117 [FREE Full text] [doi: 10.1038/sdata.2018.117] [Medline: 29917015]
10. Dugas M, Meidt A, Neuhaus P, Storck M, Varghese J. ODMedit: uniform semantic annotation for data integration in medicine based on a public metadata repository. BMC Med Res Methodol 2016 Jun 01;16:65 [FREE Full text] [doi: 10.1186/s12874-016-0164-9] [Medline: 27245222]
11. Ong TC, Kahn MG, Kwan BM, Yamashita T, Brandt E, Hosokawa P, et al. Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading. BMC Med Inform Decis Mak 2017 Sep 13;17(1):134 [FREE Full text] [doi: 10.1186/s12911-017-0532-3] [Medline: 28903729]
12. Pecoraro F, Luzi D, Ricci FL. Designing ETL tools to feed a data warehouse based on electronic healthcare record infrastructure. Stud Health Technol Inform 2015;210:929-933. [Medline: 25991292]
13. Post AR, Krc T, Rathod H, Agravat S, Mansour M, Torian W, et al. Semantic ETL into i2b2 with Eureka!. AMIA Jt Summits Transl Sci Proc 2013 Mar 18;2013:203-207 [FREE Full text] [Medline: 24303265]

14. Post AR, Pai AK, Willard R, May BJ, West AC, Agravat S, et al. Metadata-driven clinical data loading into i2b2 for Clinical and Translational Science Institutes. AMIA Jt Summits Transl Sci Proc 2016 Jul 20;2016:184-193 [FREE Full text] [Medline: 27570667]
15. Nadkarni P, Marenco L. Chapter 2 - data integration: an overview. In: Methods in Biomedical Informatics: A Pragmatic Approach. Cambridge: Academic Press; 2014.
16. Rahm E, Bernstein P. A survey of approaches to automatic schema matching. The VLDB J 2001;10:334-350. [doi: 10.1007/s007780100057]
17. Stöhr MR, Günther A, Majeed RW. Provenance for biomedical ontologies with RDF and Git. Stud Health Technol Inform 2019 Sep 03;267:230-237. [doi: 10.3233/SHTI190832] [Medline: 31483277]
18. Stöhr MR, Günther A, Majeed RW. Verifying data integration configurations for semantical correctness and completeness. Stud Health Technol Inform 2019 Sep 03;267:66-73. [doi: 10.3233/SHTI190807] [Medline: 31483256]
19. Kadioglu D, Breil B, Knell C, Lablans M, Mate S, Schlue D, et al. Samply.MDR - A metadata repository and its application in various research networks. Stud Health Technol Inform 2018;253:50-54. [Medline: 30147039]
20. Dugas M, Neuhaus P, Meidt A, Doods J, Storck M, Bruland P, et al. Portal of medical data models: information infrastructure for medical research and healthcare. Database (Oxford) 2016 Feb 11;2016:bav121 [FREE Full text] [doi: 10.1093/database/bav121] [Medline: 26868052]
21. Stöhr MR, Majeed RW, Günther A. Using RDF and Git to realize a collaborative metadata repository. Stud Health Technol Inform 2018;247:556-560. [Medline: 29678022]
22. Miller E. An introduction to the resource description framework. Bul Am Soc Inf Sci Tech 2005 Jan 31;25(1):15-19. [doi: 10.1002/bult.105]
23. Pastor-Sanchez J, Martínez-Mendez F, Rodríguez-Muñoz J. Advantages of thesaurus representation using the Simple Knowledge Organization System (SKOS) compared with proposed alternatives. Inf Res 2009 Dec;14(4) [FREE Full text]
24. Weibel SL, Koch T. The Dublin Core Metadata Initiative. D-Lib Magazine 2000 Dec;6(12) [FREE Full text] [doi: 10.1045/december2000-weibel]
25. ISO/TS 21526 Health informatics - Metadata repository requirements (MetaRep). International Organization for Standardization. 2019. URL: https://www.iso.org/standard/71041.html [accessed 2021-11-16]
26. Halilaj L, Grangel-Gonzalez I, Coskun G, Auer S. Git4Voc: Git-based versioning for collaborative vocabulary development. In: Proceedings of the 2016 IEEE Tenth International Conference on Semantic Computing (ICSC). 2016 Presented at: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC); Feb 3-5, 2016; Laguna Hills, California. [doi: 10.1109/icsc.2016.44]
27. Arndt N, Radtke N, Martin M. Distributed collaboration on RDF datasets using Git. In: Proceedings of the 12th International Conference on Semantic Systems. 2016 Presented at: SEMANTiCS 2016: 12th International Conference on Semantic Systems; Sep 12 - 15, 2016; Leipzig Germany. [doi: 10.1145/2993318.2993328]
28. SPARQL Protocol And RDF Query Language (SPARQL). W3C Semantic Web. URL: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/ [accessed 2021-11-16]
29. Stöhr MR, Majeed RW, Günther A. Metadata import from RDF to i2b2. Stud Health Technol Inform 2018;253:40-44. [Medline: 30147037]
30. Yen P, Bakken S. A comparison of usability evaluation methods: heuristic evaluation versus end-user think-aloud protocol - an example from a web-based communication tool for nurse scheduling. AMIA Annu Symp Proc 2009 Nov 14;2009:714-718 [FREE Full text] [Medline: 20351946]
31. Reen GK, Muirhead L, Langdon DW. Usability of health information websites designed for adolescents: systematic review, neurodevelopmental model, and design brief. J Med Internet Res 2019 Apr 23;21(4):e11584 [FREE Full text] [doi: 10.2196/11584] [Medline: 31012856]
32. ISO 9241-110 Ergonomics of human-system interaction - Part 110: interaction principles. International Organization for Standardization. 2020. URL: https://www.iso.org/standard/75258.html [accessed 2021-11-16]
33. Tullis T, Stetson J. A comparison of questionnaires for assessing website usability. In: Proceedings of the Usability Professionals' Association Conference. 2004 Presented at: Usability Professionals' Association Conference; Jun 7-11, 2004; Minneapolis, Minnesota, USA.
34. Brooke J. SUS: a 'Quick and Dirty' usability scale. In: Usability Evaluation In Industry. Boca Raton, Florida: CRC Press; 1996.
35. McLellan S, Muddimer A, Peres S. The effect of experience on System Usability Scale ratings. J Usability Stud 2012;7(2):56-67 [FREE Full text]
36. Bastien JC. Usability testing: a review of some methodological and technical aspects of the method. Int J Med Inform 2010 Apr;79(4):e18-e23. [doi: 10.1016/j.ijmedinf.2008.12.004] [Medline: 19345139]
37. GoToMeeting. LogMeIn. URL: https://www.gotomeeting.com/ [accessed 2021-05-07]
38. Virzi RA. Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough? Hum Factors 2016 Nov 23;34(4):457-468. [doi: 10.1177/001872089203400407]
39. Bangor A, Kortum P, Miller J. Determining what individual SUS scores mean: adding an adjective rating scale. J Usability Stud 2009;4(3):114-123.

40. Collaborative Metadata Repository (CoMetaR) Code Repository. GitHub. URL: https://github.com/dzl-dm/cometar [accessed 2021-11-16]
41. Collaborative Metadata Repository (CoMetaR) Web Application. Stöhr MR. URL: https://data.dzl.de/cometar [accessed 2021-11-16]

Abbreviations

CoMetaR: collaborative metadata repository
DC: Dublin Core
DZL: Deutsches Zentrum für Lungenforschung
FAIR: Findability, Accessibility, Interoperability, Reusability
FEV1: Forced Expiratory Volume in 1 Second
RDF: Resource Description Framework
SKOS: Simple Knowledge Organization System
SPARQL: SPARQL Protocol and Resource Description Framework Query Language

Edited by C Lovis; submitted 10.05.21; peer-reviewed by D Kadioglu, H Storf, A Blatch-Jones, C Schüttler, M Storck; comments to author 10.07.21; revised version received 13.08.21; accepted 11.10.21; published 29.11.21

Please cite as:
Stöhr MR, Günther A, Majeed RW
The Collaborative Metadata Repository (CoMetaR) Web App: Quantitative and Qualitative Usability Evaluation
JMIR Med Inform 2021;9(11):e30308
URL: https://medinform.jmir.org/2021/11/e30308
doi: 10.2196/30308
PMID:

5. Diskussion

Ein Metadatenkatalog, der alle nötigen Parameter und Variablen zur Beantwortung akuter Forschungsfragen enthält und diese wiederum durch detaillierte Annotationen unmissverständlich beschreibt, bildet die Grundlage für Kollaboration in heterogenen Forschungsverbünden. Unabdingbare Aspekte bei der Erstellung eines solchen Katalogs sind eine sichere und zuverlässige technische Infrastruktur, eine flexibel erweiterbare Repräsentation der Daten, eine den FAIR-Prinzipien entsprechend nachhaltige Formulierung und Präsentation der Metadaten, sowie eine inhaltliche Ausarbeitung unter Einbeziehung aller im Verbund verfügbarer Expertisen. Die in dieser Arbeit vorgestellte Lösung realisiert all diese Aspekte in einem modular aufgebauten Metadaten-Repository, dessen Komponenten auf international anerkannten Standards beruhen.

5.1. Technische Infrastruktur

Von technischer Seite muss gewährleistet werden, dass die Speicherung der aktuellen und aller früheren Versionen des Metadatenkatalogs, das Einpflegen neuer Versionen durch autorisierte Mitarbeiter sowie die Veräußerung von Daten den aktuellen Standards zur Informationssicherheit entsprechen. Alle vorgestellten Ergebnisse wurden auf Grundlage der in 1.4.8 beschriebenen Sicherheitsarchitektur gewonnen. CoMetaR ist unabhängig vom Betriebssystem des Servers. Der verwendete Webserver kann ebenfalls frei gewählt werden. Ähnliches gilt für den Triple Store, welcher die RDF Daten via SPARQL Schnittstelle zur Verfügung stellt. Es bleibt dem Administrator überlassen, ob Metadaten-Repository und Data Warehouse Software auf demselben Server laufen sollen oder auf getrennten. Dementsprechend bietet CoMetaR Flexibilität bei der Implementation und kann an individuelle Bedürfnisse angepasst werden. Als zentraler Bestandteil wird der Einsatz von Git erachtet, wodurch Aufgaben wie Versionierung sowie die Zusammenführung parallel erarbeiteter Versionen des Metadatenkatalogs – das sogenannte Conflict Management – bereits gelöst sind. Die Zuverlässigkeit von Git als Persistenzschicht wurde auch in der Praxis des DZL, in dem seit 2016 dasselbe Git-Verzeichnis in Nutzung ist, bestätigt. Bei Fragen zur Entwicklung einzelner Parameter konnte bisher jede einzelne Änderung seit Implementierung des Metadatenkatalogs nahtlos nachvollzogen werden. In den Größen Funktionalität, Bekanntheitsgrad und Aktivität der internationalen Community ist kein mit Git vergleichbares Instrument bekannt.

Ein möglicher nächster Schritt, um CoMetaR der internationalen Forschungsgemeinschaft zugänglicher zu machen, ist die Vereinfachung der Installation. Diese verlangt in der aktuellen Version von CoMetaR noch viele einzelne manuelle Schritte und ist dadurch komplex. Diese Komplexität kann mit Hilfe der Docker-Technologie [30], die einen fertigen virtuellen Server mit allen nötigen Diensten und Komponenten zur Verfügung stellt, deutlich reduziert werden. Hierdurch würde sich der Arbeitsaufwand der Installation von CoMetaR auf die Festlegung einiger Umgebungsvariablen beschränken.

5.2. Modellierung von Metadaten

Die Art und Weise wie Metadaten formal repräsentiert und welche Vokabularien angewandt werden, entscheidet darüber, wie zuverlässig die Informationen interpretiert und wie flexibel das Modell an sich verändernde Herausforderungen angepasst werden kann. Das Datenmodell von CoMetaR basiert auf RDF und einer Auswahl von standardisierten Vokabularien wie SKOS und Dublin Core. Diese Zusammenstellung dient einerseits der Beschreibung einzelner klinischer Parameter (beispielsweise die Einheit „Liter“ von „FEV1“) und andererseits deren semantischer Zusammenhänge (beispielsweise „FEV1 post-bronchodilatation“ als spezifischerer und deshalb untergeordneter Parameter von „FEV1“).

Manche Eigenschaften eines Parameters, wie zum Beispiel die Einheit oder der Status (draft ja/nein), sind für die Lungenforschung von hohem Interesse, können aber nach eigenen Recherchen mit den international verbreiteten Vokabularien nicht adäquat abgebildet werden. Das Modell DZL wurde deshalb um proprietäre Prädikate erweitert. Was die betroffenen Eigenschaften angeht, besteht dementsprechend eine eingeschränkte Vergleichbarkeit zu anderen Metadatensammlungen und folglich auch der hierdurch beschriebenen patientenbezogenen Instanzdaten. Eine Interpretation unseres Metadaten-Modells durch einen menschlichen Forscher mit terminologischem Hintergrundwissen wird jedoch als intuitiv möglich erachtet.

Grundsätzlich erlaubt CoMetaR durch die Verwendung von SKOS eine Abbildung jeglicher Thesauri. Gleichzeitig wird mit der Verwendung der „hasPart“- und „partOf“-Relationen, beide in RDF selbst definiert, eine weitere Möglichkeit zur Strukturierung von Metadaten geboten. Als Alternative oder auch Ergänzung zu SKOS ist an dieser Stelle die Web Ontology Language (OWL) zu nennen [31]. OWL kann ebenfalls

innerhalb von RDF formuliert werden und unterstützt nativ durch die Subklassen-Relation hierarchische Strukturen. Die Sprache bietet aber prinzipiell mehr Werkzeuge, um Metadaten und insbesondere Ontologien zu definieren. Für den Fall der Definition eines einfachen Thesaurus ist die geringere Komplexität von SKOS ein deutlicher Vorteil gegenüber OWL. SKOS wurde exakt für diesen Anwendungszweck als internationaler Standard entwickelt [14] und wird an dieser Stelle ausdrücklich empfohlen.

Was den hierarchischen Aufbau angeht, ist es dem Anwender überlassen, wie die Ordnung von Parametern untereinander zu interpretieren ist. So kann beispielsweise „Lungenkrebs“ der Diagnose „Krebs“ als Spezifizierung untergeordnet werden, in einem anderen Szenario könnte aber „Lungenkrebs“ als Kategorie einem „Anamnese-fragebogen“ untergeordnet sein. Diese Flexibilität bei der Implementierung bietet gleichzeitig Raum für mögliche Inkonsistenzen im Datenmodell. Die Bedeutung einer Unterordnung sollte innerhalb eines Metadatenkatalogs einheitlich sein. Was die Interpretation von Eigenschaften eines Parameters an sich angeht, so kann gegebenenfalls das angewandte Kodierungssystem zur korrekten Interpretation eines Wertes von Interesse sein. Beispielsweise kann „uL“ missinterpretiert werden, wenn nicht bekannt ist, dass der Wert eine Einheit aus Unified Code for Units of Measure (UCUM) [32] ist und „Mikroliter“ bedeutet. RDF bietet die Möglichkeit, Werte von Attributen näher zu definieren, indem ihr Datentyp angegeben wird. Es gilt zu prüfen, in welchen Fällen eine Angabe des Kodierungssystems als Datentyp sinnvoll ist. Im Falle des DZL existiert noch kein niedergeschriebener und veröffentlichter Leitfaden zur Modellierung von Metadaten im Kontext der Lungenforschung. Dies wird Teil zukünftiger Arbeit sein und muss unter Berücksichtigung des unter 5.3 beschriebenen Aspekts der Wiederverwendbarkeit geschehen.

RDF dient der Beschreibung von Ressourcen in Form von Eigenschaften sowie Beziehungen zu anderen Ressourcen und ist durch seine Flexibilität und Aussagekraft prädestiniert für die Repräsentation von biomedizinischen Metadaten. Grundsätzlich könnten aber statt Textdateien mit RDF-Ausdrücken auch alternative Formen der Datenhaltung wie beispielsweise eine relationale Datenbank oder ein Tabellenkalkulationsprogramm verwendet werden. Die Erweiterung und Restrukturierung einer relationalen Datenbank zur Anpassung an neue Anforderungen kann sich genau wie die Versionierung der Daten als schwierig erweisen. Microsoft Excel hätte klare Vorteile was Nutzer-Vorerfahrung angeht, eröffnet aber ganz neue

Herausforderungen was automatische Verarbeitung von Daten angeht. Für RDF wiederum spricht nicht zuletzt, dass im Jahr 2012 auch BioPortal – die größte bekannte Sammlung an Thesauri im biomedizinischen Bereich – dazu übergegangen ist, ihre Datensätze in RDF zu transformieren und via SPARQL zur Verfügung zu stellen [33].

5.3. FAIR-Prinzipien in CoMetaR

Durch die Einhaltung der FAIR-Prinzipien wird eine nachhaltige Erfassung von Metadaten sichergestellt, wodurch zukünftig die Mitwirkung in nationalen und internationalen Forschungsvorhaben ermöglicht oder erleichtert wird. Diese Arbeit hatte den Anspruch die Aspekte „Findability“, „Accessibility“, „Interoperability“ und „Reusability“ zu einem hohen Maße zu erfüllen. In den vorgestellten Manuskripten wurden Teilaspekte dieser Prinzipien behandelt. In den folgenden Abschnitten wird eine Bewertung von CoMetaR im Hinblick auf gesamtheitliche Erfüllung des Leitfadens durchgeführt.

Durch das von uns genutzt Datenmodell können die Metadaten in CoMetaR als „findable“, also auffindbar, bezeichnet werden. RDF verlangt per Definition die Angabe eines global eindeutigen Identifiers pro Ressource. Auffindbarkeit verlangt auch, dass Daten durch „rich metadata“ beschrieben sind, ohne näher zu definieren, an welche Voraussetzungen die Erfüllung des Kriteriums gebunden ist. CoMetaR verlangt als Mindestmaß die Angabe eines englischen Bezeichners, was alleinstehend Interpretationsspielraum zulassen kann. Andererseits kann rein technisch nie geprüft werden, ob die Gesamtheit beschreibender Metadaten eines Parameters diesen unmissverständlich definieren. Diese Beurteilung kann nur durch Menschen getroffen werden. Es wird empfohlen die Definition spätestens zu erweitern, wenn Verständnisschwierigkeiten aufkommen. Nicht zuletzt kann die Eindeutigkeit durch die Angabe eines internationalen Codes, für den in der Regel ausführliche Definitionen hinterlegt sind, erreicht werden. Ein standardisierter Code stellt ebenfalls ein Kriterium für auffindbare Metadaten dar und wird in CoMetaR durch das SKOS-Attribut „notation“ realisiert. Zuletzt sind die Metadaten in CoMetaR über das SPARQL-Interface oder in spezifischer Form über die Web-Applikation durchsuchbar, wodurch auch das vierte und letzte Kriterium der Auffindbarkeit erfüllt ist.

Die „Accessibility“, also Zugänglichkeit, der Metadaten wird durch eine Kombination aus HTTPS und SPARQL gewährleistet. Während über den Webserver eine

Authentifizierung und Autorisierung erfolgt, können die Annotationen einer oder mehrerer Ressourcen über die standardisierte Abfragesprache erlangt werden. Die terminologischen Metadaten sind strikt von den patientenbezogenen Instanzdaten getrennt und somit unabhängig verfügbar.

Was Interoperabilität betrifft, so erfüllt CoMetaR durch die Nutzung von RDF das Erfordernis nach einer formalen, zugänglichen und breit anwendbaren Repräsentation von Metadaten. Es bleibt allerdings der konkreten Umsetzung überlassen, ob Referenzen zu anderen Wissensdatenbanken implementiert werden. Im DZL wird angestrebt, wann immer möglich, dieselben global eindeutigen Bezeichner anerkannter Terminologien wie beispielsweise SNOMED-CT oder LOINC zu verwenden. Diese Zuordnung stellt zwar einen erheblichen Mehraufwand dar, wird aber zur Erreichung des Ziels Interoperabilität für bestehende und sich im Aufbau befindende Metadatenkataloge ausdrücklich empfohlen.

Das Prinzip „Reusability“, also Wiederverwendbarkeit, wird durch CoMetaR selbst nur bedingt erfüllt. In dieser Arbeit wurde eine Anwendung vorgestellt, die Provenance-Daten aus einem bestehenden Git Repository extrahiert und in Form von RDF-Ausdrücken zur Verfügung stellt. Allerdings ist zum Metadatenkatalog keine explizite Nutzungslizenz hinterlegt, wie es der zweite Aspekt der Wiederverwendbarkeit erwartet. Der dritte Aspekt hängt mit dem zuvor genannten Punkt der „rich metadata“ zusammen und bezieht sich auf die Erfüllung fachlich relevanter Anforderungen. Im Bereich der Biomedizin gibt es den Fast Healthcare Interoperability Resources (FHIR) Standard von Health Level 7 (HL7) International, welcher zunächst einen Kommunikationsstandard für biomedizinische Daten darstellt [34]. In diesem sind je nach Ressource Mindestanforderungen definiert. Darüber hinaus arbeitet eine 2021 gegründete Arbeitsgruppe, bestehend aus Forschern der Medizininformatik sowie Vertretern der Industrie aus ganz Deutschland, an der Erstellung eines Standards für ein Data Dictionary Minimal Information Model (DDMIM) [35]. Diese und potenziell weitere Entwicklungen werden in zukünftiger Forschungsarbeit herangezogen werden müssen, um dem Aspekt der Wiederverwendbarkeit im Sinne fachlicher Anforderungen gerecht zu werden.

5.4. CoMetaR und Datenintegration patientenbezogener Daten

Eine Verknüpfung terminologischer Metadaten mit patientenbezogenen Instanzdaten via Identifier ermöglicht Terminologie-basierte Abfragen an ein Data Warehouse sowie die

Prüfung aller der Datenintegration zugrundeliegenden Harmonisierungsregeln anhand der Annotationen der von Regeln betroffenen Parameter. Im DZL führte die Prüfung des Datenintegrationsprozesses in der Vergangenheit mehrfach zur Identifikation von falschen oder unvollständigen Regeln. Fraglich ist jedoch an dieser Stelle, ob diese Unzulänglichkeiten auch durch Prüfung des schlussendlichen Datenbestands hätten identifiziert werden können. Eine solche nachträgliche Prüfung könnte in ein Gesamtkonzept zur Bewertung der Datenqualität eingebracht werden, das weitere Aspekte wie Validität, Plausibilität und Vollständigkeit der Daten in Betracht zieht. Dies wird Teil zukünftiger Forschungsarbeit sein. Fest steht, dass auch bei einem solchen Vorhaben wohldefinierte Metadaten einen Beitrag zur Erfolgreichen Qualitätsbewertung leisten können, beispielsweise die Kontrolle, ob ein Datenbankeintrag mit korrektem Datentypen vorliegt.

Was das Tätigen von Terminologie-basierten Abfragen an ein Data Warehouse angeht, so spielt auch hier die unter 5.2 angesprochene Modellierung der Metadaten sowie deren Übersetzung auf das Datenschema des Data Warehouse eine große Rolle. Im Falle des DZL und der Nutzung von i2b2 ist das Resultat einer Abfrage immer eine Anzahl von Patienten, deren Datenbestand die formulierten Kriterien der Abfrage erfüllen. Das Kriterium des Vorhandenseins eines bestimmten Parameters wird auch dann erfüllt, wenn ein untergeordneter Parameter vorhanden ist. Diese beiden Aspekte können je nach eingesetzter Data Warehouse Software und dem definierten Modell zur Erfüllung von Kriterien variieren. Die Software i2b2 erlaubt auch virtuelle Parameter wie beispielsweise verschiedene Altersbereiche, welche sich alle auf denselben Wert, nämlich das Geburtsdatum des Patienten, beziehen. Es gilt zu prüfen, inwieweit – im DZL und in der biomedizinischen Forschung im Allgemeinen – alternative Konzepte zur Erfüllung von Abfragen den Nutzen eines Data Warehouse erhöhen. Eine mögliche Anwendung wäre die Angabe zur Berufsbezeichnung, aus welcher sich weitreichende Informationen (beispielsweise Schadstoffexpositionen) ableiten lassen, ohne diese konkret in einem Fragebogen zu erfassen.

5.5. Kollaboration

Um sich über definierte Metadaten zu informieren, diese zu diskutieren und gegebenenfalls Anpassungen vornehmen zu können, muss ein Metadaten-Repository all diese Prozesse so unterstützen, dass deren Ausübung möglichst intuitiv und unabhängig von weitreichenden technischen Kompetenzen möglich ist. CoMetaR erlaubt via Web-

Applikation die Einsicht in Metadaten, die einzelne Parameter konkret beschreiben, sie zueinander (hierarchisch) in Bezug setzen und zusätzlich die Rückverfolgung (Provenance) ihrer Entwicklung erlauben. Die grundsätzliche Aufteilung der Benutzeroberfläche in Parameterbaum auf der linken Seite und dem ausgewählten Modul (Kern, Datenintegration oder Provenance) auf der rechten Seite hat sich in der Vergangenheit als nutzerfreundlich herausgestellt (vergleiche Manuskript 7). Im selben Manuskript wurde festgestellt, dass die Aktivierung von Funktionen wie der Suche, des Datenintegrations-Moduls und des Provenance-Moduls zu einer Überladung des Parameterbaums führen kann. Zudem ist eine Erweiterung der in CoMetaR angezeigten Annotationen eines Parameters nur möglich, indem der Quellcode selbst angepasst wird. Die Darstellung von Metadaten wird also als gut, aber durchaus verbesserungswürdig angesehen.

Geplant war anfangs auch die Möglichkeit einzelne Parameter und Teilbäume in der Web-Applikation direkt zu diskutieren. Diese Überlegung wurde aufgrund des Aufwands zur Implementierung einer geeigneten Authentifizierungsmethode nicht weiterverfolgt. Eine Möglichkeit zum Single Sign-On, also einer einzigen Benutzerkennung, mit der mehrere Applikationen wie beispielsweise Metadaten-Browser und Data Warehouse genutzt werden können, wäre an dieser Stelle hilfreich gewesen. Die Implementierung eines solchen Single Sign-On wird dementsprechend vor allem dann empfohlen, wenn der Zugang zur CoMetaR Web-Applikation eingeschränkt werden soll. Was die Diskussion der Metadaten selbst anbelangt, so erfolgte diese im DZL synchron in Telefonkonferenzen sowie asynchron via digitaler Dokumente wie Text- und Excel-Dateien.

Bei der Bearbeitung von Metadaten spielt Excel insofern eine Rolle, als dass über die Jahre eine klare Präferenz gegenüber der Bearbeitung von RDF-Textdateien festgestellt wurde. Bestimmte Hürden wie Einarbeitungszeit, komplexe Syntax und aufkommende Konflikte, wenn unabhängig unterschiedliche Änderungen an den Metadaten vorgenommen wurden, konnten auch durch die in Manuskript 6 beschriebene Editor-Erweiterung nicht ausgeräumt werden. Aus diesen Hürden folgt, dass gewünschte Änderungen oftmals in Form von Excel-Sheets an das Datenmanagement des DZL kommuniziert wurden, wo daraufhin die formale Umsetzung erfolgte. Es gilt zu prüfen, inwiefern eine Bearbeitung von Metadaten in Excel-Dokumenten und die automatische Übersetzung in ein Format wie RDF realisierbar sind. Neben der Notwendigkeit einer

Struktur, die Mehrdeutigkeiten der Angaben ausschließt, wird hier auch die Wahrung der FAIR-Prinzipien eine große Rolle spielen.

5.6. Priorisierung und zeitlicher Ablauf

Der Methodik dieser Arbeit lag neben den beschriebenen Fragestellungen auch die Anforderung eines zügigen produktiven Betriebs zugrunde. So sollten einerseits die in Excel vorhandenen Metadaten mit Priorität in ein automatisch verarbeitbares Format gebracht und Nutzern übersichtlich dargestellt werden. Andererseits sollten gleichzeitig alle bekannten Register und Studiendatenbanken an das zentrale Data Warehouse angebunden und deren Daten mit korrekter Zuordnung integriert werden. Hieraus resultierte, dass zunächst das Metadaten-Repository (in Form von Git) sowie das Data Warehouse installiert wurden. Zeitgleich zur Übersetzung der vorhandenen Metadaten in RDF durch medizinische Dokumentare wurde das Kern-Modul der Web Applikation entwickelt, um die Metadaten zu visualisieren. Ab diesem Zeitpunkt konnten lokale Datenmanager Informationen zu den integrierbaren Parametern einsehen und insbesondere den Code verwenden, welcher eine Verknüpfung von terminologischen Metadaten und patientenbezogenen Daten darstellt. Den folgenden Entwicklungen, namentlich dem Datenintegrationsmodul, dem Provenance-Modul und der Editor-Erweiterung, wird ein vergleichbarer Mehrwert zugeschrieben. Sie hätten grundsätzlich in einer anderen Reihenfolge entwickelt werden können, woraus sich unseres Erachtens aber kein signifikanter Vorteil ergeben hätte.

5.7. Beitrag zum Fortschritt der Wissenschaft

Der Mehrwert dieser Arbeit besteht in der Darbietung von Instrumenten zu nachhaltigem und kollaborativem Metadaten-Management sowie der konkreten Anwendung dieser Instrumente bei der Entwicklung eines lungenspezifischen Metadatenkatalogs. Die gesamte Anwendung CoMetaR ist frei verfügbar [36]. Eine Implementierung in anderen Forschungseinrichtungen ist grundsätzlich möglich, soll aber zukünftig durch eine einfachere Installation praktikabler werden. Die zentrale IT des Deutsche Zentrum für Infektionsforschung (DZIF) nutzt seit Ende 2021 das im DZL verwendeten Setup bestehend aus CoMetaR und i2b2 im Evaluationsbetrieb. Die erfolgreiche Umsetzung von Metadaten-Repository und Data Warehouse in einem neuen Forschungskontext würde einen allgemeineren Nutzen dieser Arbeit unterstreichen.

Zur Zeit dieser Arbeit werden bereits weitere Anwendungen für eine Forschungszentrum-übergreifende Extraktion, Bereitstellung und Zusammenführung von Metadaten diskutiert. So gilt es die Metadaten aller Deutschen Zentren für Gesundheitsforschung untereinander zu vergleichen und einen minimalen Datensatz zu identifizieren, welcher von allen Zentren beziehungsweise all ihren teilnehmenden Studien, Register und Kliniken erfasst wird. Die Verfügbarkeit gut annotierter Parameter durch eine standardisierte Schnittstelle vereinfacht diesen Prozess seitens des DZL deutlich. Die bereits in 5.3 erwähnte DDMIM Initiative formuliert Mindestanforderungen an Metadatenkataloge, welche das DZL aller Voraussicht nach durch eine spezielle Abfrage an die SPARQL Schnittstelle erfüllen können wird.

Was den Fortschritt des Metadatenkatalogs im DZL selbst angeht, so wurde über die vergangenen Jahre ein erhebliches Wachstum an definierten Parametern und ihren Attributen verzeichnet. Unter Berücksichtigung der speziellen Ausrichtung auf das Feld der Lungenforschung wurde eine Terminologie geschaffen, wie sie zuvor unter Wahrung der FAIR-Prinzipien nicht existierte. Dies ermöglichte mehrfach eine Terminologie-basierte Formulierung von Abfragen an unser Data Warehouse. Hierdurch konnten bereits Forschungsvorhaben unterstützt werden, beispielsweise durch Identifikation einer spezifischen Lungenkrebskohorte [37]. Ein nächster Schritt wird sein, den entwickelten Metadatenkatalog für die internationale Forschungsgemeinschaft sichtbarer zu machen. Dieses Ziel kann beispielsweise durch die Aufnahme in eines der globalen Metadatenkatalog-Register wie FAIRsharing [38] erreicht werden.

5.8. Fazit

Die in dieser Arbeit vorgestellte Lösung für kollaboratives Metadaten-Management unterstützt biomedizinische Forschungseinrichtungen bei der Verwaltung von Metadaten unter Berücksichtigung der FAIR-Prinzipien. Der Nutzen der Software wird durch den erfolgreichen Einsatz im DZL in den vergangenen Jahren sowie den geplanten Einsatz in weiteren Forschungsverbünden deutlich. Gleichwohl ergeben sich Ansatzpunkte zur Weiterentwicklung. Besonders hervorzuheben sind an dieser Stelle die Notwendigkeiten zur vereinfachten Installation des Metadaten-Repository, zur vereinfachten Bearbeitung von Metadaten sowie zur Erstellung eines Leitfadens der Modellierung von Metadaten innerhalb des DZL.

6. Zusammenfassung

Das Deutsche Zentrum für Lungenforschung (DZL) hat es sich zum Ziel gesetzt durch translationale Forschung neue Therapien für Lungenerkrankungen zu entwickeln. Ein effektives Mittel hierfür sind Analysen auf einer breiten Basis von klinischen Daten, Biomaterialien und bildgebenden Verfahren. Die Grundlage für eine Zusammenarbeit über unterschiedliche Studien und Register hinweg, beispielsweise bei interdisziplinären Analysen, ist ein gemeinsames und eindeutiges Begriffsverständnis. Alle relevanten Begriffe eines Forschungsbereichs werden in einem Metadatenkatalog genau definiert. Diese Definitionen werden in einem Metadaten-Repository gespeichert, bearbeitet und von dort aus zur Verfügung gestellt. Zu Beginn dieser Arbeit waren alle bekannten Metadaten-Repositories entweder vom Datenmodell her für die biomedizinische Forschung ungeeignet, umständlich zu bedienen oder sie genügten nicht den international anerkannten FAIR-Prinzipien. Das im Zuge dieser Arbeit entwickelte Konzept stellt Forschern mehrere Werkzeuge zum Umgang mit Metadaten zur Verfügung. Hierzu zählt zunächst das Repository selbst, welches die Metadaten entsprechend internationaler Standards speichert, versioniert und zur Verfügung stellt. Mehrere Experten können gleichzeitig Änderungen an den in Textform vorliegenden Daten vornehmen und in das Repository einpflegen. Eine hierfür entwickelte Editor-Erweiterung unterstützt diesen Vorgang durch farbliche Hervorhebungen, Autovervollständigung und Vorab-Anzeige des resultierenden Metadaten-Baumes. Eine Web-Applikation greift direkt auf das Metadaten-Repository zu und erlaubt das Betrachten und Durchsuchen des gesamten Katalogs sowie das Zurückverfolgen aller vorangegangener Änderungen. Zudem erleichtert die Applikation die Verifikation von Regeln, welche die Integration patientenbezogener Instanzdaten in das Schema des zentralen DZL Data Warehouse i2b2 steuern. Unabhängig von der Übertragung von Patientendaten wird das Metadaten-Schema von i2b2 direkt nach dem Hochladen einer neuen Version des Metadatenkatalogs mittels eines hierfür entworfenen Transformationsalgorithmus angeglichen. Zuletzt ergab eine Evaluation zur Nutzbarkeit der genannten Web-Applikation unter Einbeziehung von zwölf Mitarbeitern des DZL mit verschiedenen Nutzerrollen und Vorkenntnissen eine gute bis sehr gute Bewertung. Potenzielle Hemmnisse zur Nutzung des Metadaten-Repository durch andere Forschungsverbünde sind die anspruchsvolle Bearbeitung der Metadaten sowie die komplexe Installation der Software. Neben dem Ausräumen dieser Hürden wird es zukünftig die Aufgabe sein, einen Modellierungsleitfaden für Metadaten

zu erstellen. Das in dieser Arbeit vorgestellte Konzept leistet einen Beitrag zu Wissenschaft und Forschung, indem es den FAIR-Prinzipien entsprechende Werkzeuge zur Modellierung und Veranschaulichung sowie zum Teilen von Metadaten zur Verfügung stellt. Eine erfolgreiche Anwendung ist ein im DZL entwickelter Lungenforschung-spezifischer Metadatenkatalog. Weitere Forschungszentren wie beispielsweise das Deutsche Zentrum für Infektionsforschung haben sich bereits dazu entschieden die vorgestellte Lösung als technische Grundlage für ihre zentrale Metadaten-Verwaltung einzusetzen.

7. Summary

The German Center for Lung Research (DZL) aims to develop new therapies for lung diseases through translational research. An effective method for this are analyses on a broad basis of clinical data, biomaterials and imaging techniques. The basis for collaboration across different studies and registries, for example in interdisciplinary analyses, is a common and clear understanding of terms. All relevant terms in a research area are precisely defined in a metadata catalog. These definitions are stored and edited in a metadata repository and shared from there. At the beginning of this work, all known metadata repositories were either unsuitable for biomedical research in terms of the data model, inconvenient to use, or did not satisfy the internationally recognized FAIR principles. The concept developed in the course of this work provides researchers with several tools for handling metadata. These include the repository itself, which stores, versionizes and provides the metadata in accordance with international standards. Several experts can work simultaneously on the text-formatted data and upload changes to the repository. An editor extension developed for this purpose supports this process with color highlighting, auto-completion, and advance display of the resulting metadata tree. A web application directly accesses the metadata repository and allows viewing and searching the entire catalog as well as tracing all previous changes. In addition, the application facilitates the verification of rules that control the integration of patient-related instance data into the schema of the central DZL Data Warehouse i2b2. Regardless of the transfer of patient data, i2b2's metadata schema is adapted immediately after uploading a new version of the metadata catalog using a transformation algorithm that was developed for this purpose. Lastly, an evaluation on the usability of the mentioned web application involving twelve DZL staff members with different user roles and backgrounds resulted in a good to very good rating. Potential hurdles to the use of CoMetaR by other research collaborations include the demanding processing of metadata and the complex installation of the software. In addition to removing these barriers, a future task will be to create a modeling guide for metadata in CoMetaR. CoMetaR contributes to science and research by providing tools for modeling and visualizing as well as sharing metadata in accordance with FAIR principles. One successful application is a lung research-specific metadata catalog developed at DZL. Other research centers such as the German Center for Infection Research have already decided to use the presented solution as a technical basis for their central metadata management.

8. Abkürzungsverzeichnis

ARCN ..... *Airway Research Center North*
BIC ..... *Broad Informed Consent*
BMBF ..... *Bundesministerium für Bildung und Forschung*
BREATH .... *Biomedical Research in Endstage and Obstructive Lung Disease Hannover*
BSI ..... *Bundesamts für Sicherheit in der Informationstechnik*
CDM ..... *Common Data Model*
CoMetaR ..... *Collaborative Metadata Repository*
CPC-M ..... *Comprehensive Pneumology Center Munich*
DC ..... *Dublin Core*
DDMIM ..... *Data Dictionary Minimal Information Model*
DKTK ..... *Deutsche Konsortium für Translationale Krebsforschung*
DZG ..... *Deutsches Zentrum für Gesundheitsforschung*
DZIF ..... *Deutsche Zentrum für Infektionsforschung*
DZL ..... *Deutsches Zentrum für Lungenforschung*
FAIR ..... *Findability, Accessibility, Interoperability, and Reuse*
FEV1 ..... *Forced Expiratory Volume in 1 second*
FHIR ..... *Fast Healthcare Interoperability Resources*
FVC ..... *Forced Vital Capacity*
GBN ..... *German Biobank Node*
HL7 ..... *Health Level 7*
HTTP ..... *HyperText Transfer Protocol*
HTTPS ..... *HyperText Transfer Protocol Secure*
i2b2 ..... *Informatics for Integrating Biology and the Bedside*
ICD-10 ..... *International Classification of Diseases 10th Revision*
ISO ..... *International Organization for Standardization*
JMIR ..... *Journal of Medical Internet Research*
KRITIS ..... *Kritische Infrastruktur*
LOINC ..... *Logical Observation Identifiers Names and Codes*
ODM ..... *Operational Data Model*
OMOP ..... *Observational Medical Outcomes Partnership*
OWL ..... *Web Ontology Language*
RDF ..... *Resource Description Framework*

SKOS ..... *Simple Knowledge Organization System*
SNOMED-CT ..... *Systematized Nomenclature of Medicine Clinical Terms*
SPARQL ..... *SPARQL Protocol And RDF Query Language*
SQL ..... *Structured Query Language*
TLRC ..... *Translational Lung Research Center*
TLS ..... *Transport Layer Security*
TMF ..... *Technologie- und Methodenplattform für die vernetzte medizinische Forschung*
UCUM ..... *Unified Code for Units of Measure*
UGMLC ..... *Universities of Giessen and Marburg Lung Center*
UMLS ..... *Unified Medical Language System*
USA ..... *United States of America*
W3C ..... *World Wide Web Consortium*
WHO ..... *World Health Organization*
XML ..... *Extensible Markup Language*

9. Abbildungsverzeichnis

10. Literaturverzeichnis

1. Hasenfuß G. Deutsche Zentren der Gesundheitsforschung - Schneller von der Idee zum Patienten. [German Centres for health science - faster from the idea to the patient]. Dtsch Med Wochenschr 2014;139(49):2487-2488. PMID:25423453
2. Ulrich H, Kock-Schoppenhauer A-K, Deppenwiese N, Gött R, Kern J, Lablans M, Majeed RW, Stöhr MR, Stausberg J, Varghese J, Dugas M, Ingenerf J. Understanding the Nature of Metadata: Systematic Review. J Med Internet Res 2022;24(1):e25440. PMID:35014967
3. Gonçalves RS, Musen MA. The variable quality of metadata about biological samples used in biomedical experiments. Sci Data 2019;6:190021. PMID:30778255
4. Bodenreider O, Cornet R, Vreeman DJ. Recent Developments in Clinical Terminologies - SNOMED CT, LOINC, and RxNorm. Yearb Med Inform 2018;27(1):129-139. PMID:30157516
5. Marienfeld F, Schieferdecker I, Lapi E, Tcholtchev N. Metadata aggregation at GovData.de. WikiSym '13 Proceedings of the 9th International Symposium on Open Collaboration;2013:1-5. doi:10.1145/2491055.2491077
6. Elias M, Shahzad K, Johannesson P. A Business Process Metadata Model for a Process Model Repository. In: van der Aalst W, Mylopoulos J, Sadeh NM, Shaw MJ, Szyperski C, Bider I, Halpin T, Krogstie J, Nurcan S, Proper E, Schmidt R, Ukor R, editors. Enterprise, Business-Process and Information Systems Modeling. Vol. 50. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. ISBN:978-3-642-13050-2. p. 287–300.
7. Nadkarni PM, Marenco LN. Chapter 2 - Data Integration: An Overview. Methods in Biomedical Informatics: A Pragmatic Approach 2013:15-47.
8. World Wide Web Consortium. Resource Description Framework 2014 URL: https://www.w3.org/RDF/ [accessed 2022-03-09].
9. W3C Semantic Web. Turtle – Terse RDF Triple Language 2014 URL: https://www.w3.org/TR/turtle/ [accessed 2022-03-09].
10. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ,

Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJG, Groth P, Goble C, Grethe JS, Heringa J, Hoen PAC 't, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone S-A, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 2016;3. PMID:26978244

11. International Standards for Business, Government and Society. ISO/IEC FCD 11179-3: Information technology - Metadata registries (MDR) - Part 3: Registry metamodel and basic attributes.
12. Kadioglu D, Breil B, Knell C, Lablans M, Mate S, Schlue D, Serve H, Storf H, Ückert F, Wagner T, Weingardt P, Prokosch H-U. Samply.MDR - A Metadata Repository and Its Application in Various Research Networks. Stud Health Technol Inform 2018;253:50-54. PMID:30147039
13. Stöhr MR, Günther A, Majeed RW. ISO 21526 Conform Metadata Editor for FAIR Unicode SKOS Thesauri. Stud Health Technol Inform 2021;278:94-100. PMID:34042881
14. Alistair Miles, Brian Matthews, Michael Wilson, Dan Brickley. SKOS Core: Simple knowledge organisation for the Web. International Conference on Dublin Core and Metadata Applications 2005:3-10.
15. Dublin Core Metadata Initiative. Dublin Core 2017 URL: http://dublincore.org/ [accessed 2022-09-03].
16. Clinical Data Interchange Standards Consortium. ODM-XML 2013 URL: https://www.cdisc.org/standards/data-exchange/odm [accessed 2022-02-09].
17. World Health Organization. ICD-10-WHO 2019 URL: https://www.bfarm.de/DE/Kodiersysteme/Klassifikationen/ICD/ICD-10-WHO/_node.html [accessed 2022-02-09].
18. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res 2004;32(Database issue):D267-70. PMID:14681409

19. Horridge M, Tudorache T, Nuylas C, Vendetti J, Noy NF, Musen MA. WebProtege: a collaborative Web-based platform for editing biomedical ontologies. Bioinformatics 2014;30(16):2384-2385. PMID:24771560

20. Noy NF, Shah NH, Whetzel PL, Dai B, Dorf M, Griffith N, Jonquet C, Rubin DL, Storey M-A, Chute CG, Musen MA. BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res 2009(37):170-173. PMID:19483092

21. Joos S, Nettelbeck DM, Reil-Held A, Engelmann K, Moosmann A, Eggert A, Hiddemann W, Krause M, Peters C, Schuler M, Schulze-Osthoff K, Serve H, Wick W, Puchta J, Baumann M. German Cancer Consortium (DKTK) - A national consortium for translational cancer research. Mol Oncol 2019;13(3):535-542. PMID:30561127

22. Lablans M, Kadioglu D, Mate S, Leb I, Prokosch H-U, Ückert F. Strategien zur Vernetzung von Biobanken. Klassifizierung verschiedener Ansätze zur Probensuche und Ausblick auf die Zukunft in der BBMRI-ERIC. [Strategies for biobank networks. Classification of different approaches for locating samples and an outlook on the future within the BBMRI-ERIC]. Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz 2016;59(3):373-378. PMID:26753865

23. Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, Kohane I. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc 2010;17(2):124-130. PMID:20190053

24. Overhage JM, Ryan PB, Reich CG, Hartzema AG, Stang PE. Validation of a common data model for active safety surveillance research. J Am Med Inform Assoc 2012;19(1):54-60. PMID:22037893

25. Majeed RW, Fischer P, Günther A. Accessing OMOP Common Data Model Repositories with the i2b2 Webclient - Algorithm for Automatic Query Translation. Stud Health Technol Inform 2021;278:251-259. PMID:34042902

26. Hummel M, Stege A. Der Aufbau und Betrieb einer Zentralen Biomaterialbank : Die ZeBanC der Charité Berlin. [The construction and operation of a central biomaterial bank : The ZeBanC of the Charité Berlin]. Pathologe 2018;39(4):313-319. PMID:29922857

27. Bundesamt für Sicherheit in der Informationstechnik. IT Grundschutz 2021 URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html [accessed 2022-02-21].

28. Chacon S. Pro Git. 2nd ed. Berkeley, CA: Apress L. P; 2014. ISBN:9781484200766.

29. W3C Semantic Web. SPARQL Protocol And RDF Query Language (SPARQL) 2013 URL: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/ [accessed 2020-07-14].

30. Anderson C. Docker [Software engineering]. IEEE Softw. 2015;32(3):102-c3. doi:10.1109/ms.2015.62

31. ANTONIOU G, van Harmelen F. Web Ontology Language: OWL:67-92. doi:10.1007/978-3-540-24750-0_4

32. Schadow G, McDonald CJ. The Unified Code for Units of Measure 1999 URL: https://ucum.org/ [accessed 2022-02-28].

33. Salvadores M, Horridge M, Alexander PR, Fergerson RW, Musen MA, Noy NF. Using SPARQL to Query BioPortal Ontologies and Metadata. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Cudré-Mauroux P, Heflin J, Sirin E, Tudorache T, Euzenat J, Hauswirth M, Parreira JX, Hendler J, Schreiber G, Bernstein A, Blomqvist E, editors. The Semantic Web – ISWC 2012. Vol. 7650. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. ISBN:978-3-642-35172-3. p. 180–195.

34. Bender D, Sartipi K. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. doi:10.1109/cbms.2013.6627810

35. Kadioglu D, Löbe M, Stöhr MR, Vengadeswaran A, Majeed RW. Towards a Data Dictionary Minimal Information Model – Consensus for Research Metadata Exchange. 66. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e. V. (GMDS), 12. Jahreskongress der Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V. (TMF). [German Medical Science GMS Publishing House] 2021. doi:10.3205/21GMDS030

36. Stöhr MR. Collaborative Metadata Repository (CoMetaR) Code Repository URL: https://github.com/dzl-dm/cometar [accessed 2022-03-09].

37. Walter J, Kauffmann-Guerrero D, Muley T, Reck M, Fuge J, Günther A, Majeed R, Dinkel J, Schneider C, Senghas K, Watermann I, Kobinger S, Koch I, Manapov F, Thomas M, Kahnert K, Winter H, Behr J, Tammemagi M, Tufman A. P61.03 Comparison of the Sensitivity of Different Screening Algorithms to Select Lung Cancer Patients for Screening in a Cohort of German Patients. Journal of Thoracic Oncology 2021;16(10):S1172-S1173. doi:10.1016/j.jtho.2021.08.638

38. Sansone S-A, McQuilton P, Rocca-Serra P, Gonzalez-Beltran A, Izzo M, Lister AL, Thurston M. FAIRsharing as a community approach to standards, repositories and policies. Nat Biotechnol 2019;37(4):358-367. PMID:30940948

11. Anhänge

Die folgenden Seiten enthalten die 2017 von der „DZL Platform Biobanking and Data Management“ konsentierten Listen zur Kategorisierung und Kodierung aller bis dato für die Lungenforschung als relevant erachteter Spezimen und Phänotypen.

Blood and derivates / Blut und Derivate Präfix: B-

Blood products	Blutprodukte	Neuer Code	Alter Code	SPREC 2
Whole Blood	Vollblut	W	BLU_999	BLD
EDTA	EDTA	WE	BLU_001	
Nucleic acid stabilized	Nukleinsäure Stabilisierung	WNA	BLU_002	
PAXgene	PAXgene	WNAP		PAX
Tempus Tube	Tempus Tube	WNAT		TEM
Streck Tube	Streck tube	WNAS		
Plasma	Plasma	PL		PL1
Citrated	Citrat	PLC	BLU_003	
Heparinized	Heparin	PLH	BLU_004	
EDTA	EDTA	PLE	BLU_005	
Protease Inhibitor	Proteaseinhibitor	PLPI	BLU_006	PIX
Serum	Serum	SE	BLU_007	SER
Gel	Gel	SEG		
Pellet	Pellet	SEP		

Derivate

Others	Sonstige	Neuer Code	Alter Code	SPREC 2.0
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele:

Blood and Derivates	Neuer Code	Alter Code	SPREC 2.0
EDTA whole Blood	B-WE	BLU_001	
Whole Blood -RNA stabilized (Tempus tube)	B-WNAT		BLD-TEM
Plasma citrated	B-PLC	BLU_003	
Plasma heparinized	B-PLH	BLU_004	
Plasma EDTA	B-PLE	BLU_005	
Plasma Protease inhibitor (P100) stabilized	B-PLPI	BLU_006	PL1-PIX
Serum	B-SE	BLU_007	SER

Swabs and Brushings / Abstriche und Bürstungen Präfix: SB-

Extraction types	Entnahmeart	Code	SPREC 2.0
Swabs	Abstrich	SW	SWAB
Brushing	Bürstungen	BRU	

Localisation	Lokalisation	Code	SPREC 2.0
Bronchus	Bronchus	BRO	
Nose	Nase	NO	
Pharynx	Rachen	PH	
Nasopharynx	Nasenrachen	NOPH	
Nasal Polyps	Nasenpolypen	NOPO	
Rectum	Rektum	REC	
Skin	Haut	SK	

Stabil./Fix.	Stabil./Fixierung	Code	SPREC 2.0
native	nativ	FN	
Additiv	Zusatz	FA	ADD
Virom	Virom	FAV	
Microbiom	Mocrobiom	FAMI	

Derivate

Others	Sonstige	Neuer Code	Alter Code	SPREC 2.0
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele

Swabs and Brushing	Neuer Code	SPREC 2.0	Alter Code
Rectoanal Mucosal Swab	SB-SW-REC-FN		BRU_003
Pharyngeal Swab	SB-SW-PH-FN		BRU_004
Nasopharyngeal Swab	SB-SW-NOPH-FN		BRU_005
Nasal Swab	SB-SW-NO-FN		BRU_006
Nasal Brushing	SB-BRU-NO-FN		BRU_008
Nasal Polyps	SB-NOPO		BRU_009

Lavages, Airway Aspirates, Sputum/Lavage, Sekrete,Sputen — Präfix: LAS-

Extraction	Gewinnung + Ort	Code	SPREC 2.0	Kommentare:
Lavage	Lavage	L		
BAL	BAL	LB		
BAL Supernatant	BAL Überstand	LBS		
BAL Precipiate	BAL Präzipitat			see Isol. Cells
Pharynx	Rachen	LPH		
Nose	Nase	LNO		
Sputum	Sputum	SP	SPT	
Spontaneuos	Spontan	SPS		
Induced	Induziert	SPI		
Secretion	Sekrete	SE		
Trachea	Trachea	SET		
Supernatant	Überstand	SETS		
Precipitate	Präzipitat			see Isol. Cells
Nose	Nase	SENO		
EBC	EBC	EBC		
Saliva	Speichel	SAL	SAL	

Stabil./Fix.	Stabil./Fix.	Code	SPREC 2.0
Native	Nativ	FN	
Additive	Zusatz	FA	ADD
EDTA	EDTA	FAE	
BHT	BHT	FABHT	
Citrate	Citrat	FAC	
Proteaseinhibitor	Proteaseinh	FAPI	PIX

Derivate

Others	Sonstige	Neuer Code	Alter Code	SPREC 2.0
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele

Lavage, Airway Aspirates, Sputum	Neuer Code	SPREC 2.0	Alter Code
BAL Supernatant, native	LAS-LBS-FN		LAV_001
BAL Supernatant + Additive (EDTA)	LAS-LBS-FAE		
BAL Supernatant + Additive (BHT)	LAS-LBS-FABHT		
BAL Supernatant + Additive (Citrate)	LAS-LBS-FAC		
Tracheal Secretion	LAS-SET-FN		LAV_004
Pharyngeal Lavage	LAS-LPH-FN		LAV_005
Nasal Lavage	LAS-LNO-FN		LAV_006
Exhaled Breath Condensate (EBC)	LAS-EBC		LAV_007
spontaneous Sputum (supernatant), native	LAS-SPS-FN		LAV_008
spontaneous Sputum (supernatant), with Proteasei	LAS-SPS-FAPI		LAV_009
induced Sputum (supernatant), native	LAS-SPI-FN		LAV_010
induced Sputum (supernatant), with Proteaseinhibi	LAS-SPI-FAPI		LAV_011

Tissue/Gewebe Präfix: T-

Tissue type	Gewebeart	Code	SPREC 2.0
Tumor Tissue	Tumorgewebe	TT	
Lung Tissue Tumor Margin	Lungengewebe Tumorrand	LTTM	
Lung Tissue Peripheral	Peripheres Lungengewebe	LTP	
Pulmonary Artery	Pulmonalarterie	PA	
Bronchus	Bronchus	BRO	
Lymph Node	Lymphknoten	LN	
Fat Tissue	Fettgewebe	FT	
Diaphragm	Zwerchfell	DIA	
Adrenal Gland	Nebenniere	AG	
Pleura	Pleura	PLE	

Extraction	Entnahmeart	Code	SPREC 2.0
post LTX	nach LTX	PTX	
OP/resected tissue	OP/Resektat	OP/R	SRG
Biopsy	Biopsie	BX	BPS
needle	Nadel	BXN	FNA
forceps	Zange	BXF	
cryo	Cryo	BXC	

Fixation	Fixierung	Code	SPREC 2.0
Freezing	Gefrieren	F	
Native	nativ	FN	
Additive	Zusatz	FA	ADD
RNA later	RNA later	FAR	RNL
Cryoprotective	Kryokonserv.	FACR	
Paraffin embeeded	Paraffin eingebettet	P	P
FF	FF	PF	
HOPE-fixed	HOPE-fixed	PH	

Derrivatives	Derrivate	Neuer Code	Alter Code	SPREC 2.0
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele	Neuer Code	SPREC 2.0	Alter Code
Lung tissue postLTx(fresh frozen)	T-LTTM-PTX-F		TIS_001
Lung Tissue – Resection/Surgical Specimen (fresh froze	T-LTP-OP/R-F		TIS_002
Lung Tissue – Biopsy (needle) (fresh frozen)	T-LTP-BXN-F		TIS_003
Lung Tissue – Biopsy (cryo) (fresh frozen)	T-LTP-BXC-F		TIS_003
Lung Tissue – Biopsy (forceps) (fresh frozen)	T-LTP-BXF-F		TIS_003
Tumor tissue (fresh frozen)	T-TT-F		TIS_004
Lung Tissue Tumor Margin (fresh frozen)	T-LTTM-F		TIS_010
Lung Tissue Tumor Margin - RNA later	T-LTTM-FAR		TIS_013
Tumor Tissue - RNA later	T-TT-FAR		TIS_014
Pulmonary Artery	T-PA		TIS_005

Cells/Zellen Präfix: C-

Origin	Herkunft	Code neu	Code alt	SPREC 2
Tissue	Gewebe	T		TIS
Lung Tissue Common	Allgemeines Lungengewebe	TLC		
Pulmonary Artery SMC	Glatte Gefäßmuskelzellen	TSM	CEL_001	
Alveolar Type I Cells	Alveolar Typ I Zellen	TAE1	CEL_002	
Alveolar Type II Cells	Alveolar Typ II Zellen	TAE2	CEL_003	
Bronchial Epithelial Cells	Bronchialepithelzellen	TBE	CEL_004	
Vascular Endothelial Cells	Gefäßendothelzellen	TVEN	CEL_005	
Interstitial Fibroblasts	Interstitielle Fibroblasten	TIF	CEL_006	
Pulm. Art. Adventitial Fibrobl.	Adventitia-Fibroblasten	TAF	CEL_007	
Lung Tissue Single Cell Prep.	Gewebe Einzelzellpräparation (Ce	TCS	CEL_008	
Tumor Cells	Tumorzellen	TTC	CEL_009	
Liquids	Flüssigkeiten	L		
Cells from blood	Zellen aus Blut	LB		
Cell Pellet	Zellpellet	LBP		
EDTA-blood	EDTA-Blut	LBP1	BLU_009	
Heparin-blood	Heparin-Blut	LBP2	BLU_010	
TrueCulture-blood	TrueCulture-Blut	LBP3		
Leucocytes	Leukozyten	LBLE		
Buffy coat	Buffy coat	LBLE1	BLU_008	
TrueCulture	TrueCulture	LBLE2	BLU_012	
Mononuclear cells (MNC)	Mononukl. Zellen	LBMNC	CEL_010	
Ficoll	Ficoll	LBMNC1	BLU_014	
CPT	CPT	LBMNC2		
Monocytes	Monozyten	LBM		
Granulocytes	Granulozyten	LBG		
Eosinophiles	Eosinophile	LBGE	CEL_019	
Neutrophiles	Neutrophile	LBGN	CEL_011	
Lymphocytes	Lymphozyten	LBLY	CEL_012	
CD4+	CD4+	LBLY1	CEL_020	
CD8+	CD8+	LBLY2	CEL_021	
NK	NK	LBLY3	CEL_022	
B	B	LBLY4	CEL_035	
ILC	ILC	LBLY5	CEL_023	
Myeloid-derived Suppressor Cells	Myeloide Suppressorzellen (M	LBMDSC		
CD66b positive	CD66b positive	LBMDSC1		
CD66b negative	CD66b negative	LBMDSC2		
BAL Cells	BAL Zellen	LBA		
Whole	Gesamt	LBAW	CEL_039	
Alveolar Macrophages	Alveolarmakrophagen	LBAM		
Sorted	Gesortet	LBAMS	CEL_036	
Resident	Residente	LBAMR	CEL_025	
Ecsudate	Exsudat	LBAME	CEL_026	
Mononuclear cells (MNC)	Mononukl. Zellen (Monoz.+Ly	LBAMNC	CEL_038	
Monocytes	Monozyten	LBAMO	CEL_027	
Granulocytes	Granulozyten	LBAG		
Eosinophiles	Eosinophile	LBAGE	CEL_028	

Neutrophiles	Neutrophile	LBAGN	CEL_040
Sorted/FACS	Gesortet/FACS	LBAGNS	CEL_041
Lymphocytes	Lymphozyten	LBAL	CEL_029
CD4+	CD4+	LBAL1	CEL_030
CD8+	CD8+	LBAL2	CEL_031
NK	NK	LBAL3	CEL_032
B	B	LBAL4	CEL_033
ILC	ILC	LBAL5	CEL_034
Dendritic Cells	Dendritische Zellen	LBAD	CEL_035
Sputum Cells	Sputumzellen	LSP	CEL_014
Nasal Epithelial Cells	Nasenepithelzellen	LNO	CEL_013
Tracheal Secretion Cells	Trachealsekret Zellen	LSET	LAV_004
Precipitate	Präzipitat	LSETP	

Fixation	**Fixierung**	**Code**	**SPREC 2.0**
native	nativ	FN	
Additives	Zusatz	FA	
RLT Buffer	RLT Puffer	FARLT	
Medium	Medium	FAME	
RNA later	RNA later	FARC	

Type of extracti	**Art der Gewinnung**	**Code**	**SPREC 2.0**
Sorted	Gesortet	SO	
Other procedur	Andere Verfahren (Pa	OTH	

Derrivatives	**Derrivate**	**Code**	**Code alt**	**SPREC 2.0**
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele:

Cells/Zellen	**Neuer Code**	**SPREC 2.0**	**Alter Code**
isolated Cells - Pulmonary Artery Smooth Muscle Cells (S	C-TSM		CEL_001
isolated Cells - Alveolar Type I Cells (AECI)	C-TAE1		CEL_002
isolated Cells - Alveolar Type II Cells (AECII)	C-TAE2		CEL_003
isolated Cells - Bronchial Epithelial Cells	C-TBE		CEL_004
isolated Cells - single cell preparation (cell slurry)	C-TCS		CEL_008
isolated Cells - Nasal Epithelial Cells	C-LNO		CEL_013
Sputum Cells	C-LSP		CEL_014
BALF Cells - total cells	C-LBAW		CEL_039
BALF Cells - total cells with additives (RLT)	C-LBAW-FARLT		CEL_037
BALF Cells - Alveolar Macrophages (whole)	C-LBAM		CEL_015
BALF Cells - resident Alveolar Macrophages	C-LBAMR		CEL_025
BALF Cells - exsudate Alveolar Macrophages	C-LBAME		CEL_026

Blood and Derivates	**Neuer Code**	**SPREC 2.0**	**Alter Code**
Buffy coat	C-LBLE1		BLU_008
Ficoll - cells	C-LBMNC1		BLU_014
EDTA-blood Cell pellet	C-LBP1		BLU_009
TruCulture-blood Cell pellet	C-LBP3		BLU_012

Urine / Urin — Präfix: U-

Removal-Time	Entnahme-Zeit	Code	SPREC 2.0
Urine	Urin	U	
First morning	Morgenurin	U1	URM
Random ("spot")	Urin spot	U2	URN
24 h urine	Sammelurin	U3	U24

Product	Produkt	Code	Alter Code
Urine	Urin		
Midstream	Mittelstrahl	MID	LIQ_001
Precipitate	Präzipitat	PRE	LIQ _002
Supernatant	Überstand	SUP	LIQ _003

Derivate

Others	Sonstige	Neuer Code	Alter Code	SPREC 2.0
DNA	DNA	D	CEL_017	
RNA	RNA	R		
cDNA	cDNA	cD		

Beispiele

Urine	Neuer Code	Alter Code	SPREC 2.0
Urine, midstream	U-MID	LIQ_001	
Urine, precipitate	U-PRE	LIQ_002	
Urine, supernatant	U-SUP	LIQ_003	

Others/Sonstige — Präfix: O-

Others	Sonstige	Neuer Code	Alter Code	SPREC 2.0
Bacterial isolates	Bakterienis	BIS	CEL_016	
Stool sample	Stuhlprobe	STL	OTH_001	STL
Hair	Haare	HAI		
Fingernails	Fingernäge	FINA		

Phenotype Lung Disease/Phänotypen Lungenerkrankungen		Code neu	Code alt
Asthma and Allergy/Asthma und Allergie		AA	AAA_999
Adult	Adult	AAA	
allergic	allergisch	AAA-A	
mild-to-moderate	mildes-moderates	AAA-A-M	AAA_001
severe	schweres	AAA-A-S	AAA_002
non-allergic	nicht allergisch	AAA-N	
mild-to-moderate	mildes-moderates	AAA-N-M	AAA_003
severe	schweres	AAA-N-S	AAA_004
Mixed forms	Mischformen	AAA-M	
Pediatric	Pädiatrisch	AAP	
atopic	atopisch	AAP-AT	
new-onset	new-onset	AAP-AT-NO	
viral wheeze	viral wheeze	AAP-AT-NO1	
multiple-trigger	multiple-trigger	AAP-AT-NO2	
Asthma	Asthma	AAP-AT-NO3	
established	established	AAP-AT-E	
viral wheeze	viral wheeze	AAP-AT-E1	
multiple-trigger	multiple-trigger	AAP-AT-E2	
Asthma	Asthma	AAP-AT-E3	
non-atopic	nicht -atopisch	AAP-NA	
new-onset	new-onset	AAP-NA-NO	
viral wheeze	viral wheeze	AAP-NA-NO1	
multiple-trigger	multiple-trigger	AAP-NA-NO2	
Asthma	Asthma	AAP-NA-NO3	
established	established	AAP-NA-E	
viral wheeze	viral wheeze	AAP-NA-E1	
multiple-trigger	multiple-trigger	AAP-NA-E2	
Asthma	Asthma	AAP-NA-E3	
Chronic Obstructive Pulmonary Disease, COPD		COPD	COPD_999
GOLD I	GOLD I	COPD-1	COPD_001
A	A	COPD-1A	
B	B	COPD-1B	
GOLD II	GOLD II	COPD-2	COPD_002
A	A	COPD-2A	
B	B	COPD-2B	
GOLD III	GOLD III	COPD-3	COPD_003
C	C	COPD-3C	
D	D	COPD-3D	
GOLD IV	GOLD IV	COPD-4	COPD_004
C	C	COPD-4C	
D	D	COPD-4D	
Emphysema in Alpha-1-Antitrypsin Deficiency	Emphysem bei Alpha-1-Antitrypsi	COPD-A1	COPD_005
Cystic Fibrosis, CF / Mukoviszidose		CF	CF_999
PI-CF	PI-CF	CF-PI	

PI-CF (Pancreas insuff.) > 80% FEV1 % pred.	PI-CF (Pankreasinsuffizient) >	CF-PI-1	CF_001
PI-CF (Pancreas insuff.) , 60 - 80% FEV1 % pred.	PI-CF (Pankreasinsuffizient) ,	CF-PI-2	CF_002
PI-CF (Pancreas insuff.) , 40 - 60 % FEV1 % pred.	PI-CF (Pankreasinsuffizient) ,	CF-PI-3	CF_003
PI-CF (Pancreas insuff.) < 40% FEV1 % pred.	PI-CF (Pankreasinsuffizient) <	CF-PI-4	CF_004
PS-CF	PS-CF	CF-PS	
PS-CF (Pancreas suff.) > 80% FEV1 % pred.	PS-CF (Pankreassuffizient) >	CF-PS-1	CF_005
PS-CF (Pancreas suff.) , 60 - 80% FEV1 % pred.	PS-CF (Pankreassuffizient) ,	CF-PS-2	CF_006
PS-CF (Pancreas suff.) , 40 - 60 % FEV1 % pred.	PS-CF (Pankreassuffizient) ,	CF-PS-3	CF_007
PS-CF (Pankreas sufficient) < 40% FEV1 % pred.	PS-CF (Pankreassuffizient) <	CF-PS-4	CF_008
CF-related disorder (CFRD)	CF-related disorder (CFRD)	CF-RD	CF_009

Diffuse Parenchymal Lung Disease, DPLD		DP	DPLD_999
Idiopatic	Idiopatische	DP-ID	
IPF/UIP	IPF/UIP	DP-ID-IPF	DPLD_001
NSIP	NSIP	DP-ID-NSIP	DPLD_002
DIP	DIP	DP-ID-DIP	DPLD_003
RB-ILD	RB-ILD	DP-ID-RB	DPLD_004
COP/BOOP	COP/BOOP	DP-ID-COP	DPLD_005
LIP	LIP	DP-ID-LIP	DPLD_006
AIP	AIP	DP-ID-AIP	DPLD_007
AFOP	AFOP	DP-ID-AFOP	DPLD_025
IPAF	IPAF	DP-ID-IPAF	
Collagenosis	Kollagenosen	DP-KO	DPLD_010
Systemic Lupus Erythematosus (SLE)	Systemischer Lupus Erythem	DP-KO1	
Scleroderma	Sklerodermie	DP-KO2	
Rheumatoid Arthritis (RA)	Rheumatoide Arthritis (RA)	DP-KO3	
Polymyositis/Dermatomyositis	Polymyositis/Dermatomyositis	DP-KO4	
Mixed Connective Tissue Dis. (MCTD)	Mischkollagenosen (MCTD)	DP-KO5	
Spondylitis	Spondylitis	DP-KO6	
Sjögren-Syndrome	Sjögren-Syndrom	DP-KO7	
Vasculitis	Vaskulitiden	DP-VA	DPLD_012
Wegener's granulomatosis	Wegener-Granulomatose	DP-VA1	
Churg-Strauss Syndrome	Churg-Strauss Syndrom	DP-VA2	
Panarteritis nodosa	Panarteritis nodosa	DP-VA3	
Iatrogenic	Iatrogen	DP-IAT	DPLD_013
Antibiotics (Nitrofurantoin)	Antibiotika(Nitrofurantoin)	DP-IAT1	
Antiarrhythmics (Amiodarone)	Antiarrhytmika (Amiodaron)	DP-IAT2	
Chemotherapeutics (Bleomycin)	Chemotherapeutika (Bleomyc	DP-IAT3	
Radiation-Induced DPLD	Strahleninduzierte DPLD	DP-RI	DPLD_014
Environmental organic toxicants	Umweltbedingt organische Noxen	DP-OT	
Hypersensitivity pneumonitis	Exogen allergische Alveolitis	DP-EAA	DPLD_009
Environmental anorganic toxicants	Umweltbedingt anorganische Nox	DP-AT	
Pneumoconiosis	Pneumokoniosen	DP-PNE	DPLD_011
Asbestosis	Asbestose	DP-PNE1	
Silicosis	Silikose	DP-PNE2	
Berylliosis	Berylliose	DP-PNE3	
Siderosis	Siderose	DP-PNE4	
Sarcoidosis	Sarkoidose	DP-SAR	DPLD_008
Lysosomal Storage disorders	Lysosomale Speichererkrankunge	DP-LS	DPLD_015

Hermansky-Pudlak Syndrome	Hermansky-Pudlak Syndrom	DP-LS1	
Niemann-Pick Disease	M. Niemann-Pick	DP-LS2	
Morbus Gaucher	M. Gaucher	DP-LS3	
Post ARDS	Post ARDS	DP-ARDS	DPLD_016
Neurofibromatosis	Neurofibromatose	DP-NF	DPLP_017
Lymphangioleiomyomatosis	Lymphangioleiomyomatose	DP-LA	DPLD_019
Chronic inflammatory bowel disease	Chronisch entzündliche Darmerkr	DP-IBD	DPLD_018
Adult Histiocytosis X	Adulte Histiozytosis X	DP-LCH	DPLD_020
Emphysema	Emphysem	DP-EMP	DPLD_027
CPFE-Fibroemphysema	CPFE -Fibroemphysem	DP-FE	DPLD_026
DPLD-Unknown cause	DPLD-Unklare Genese	DP-UG	
adult	adulte	DP-UGA	DPLD_022
pediatric	pädiatrische	DP-UGP	DPLD_023
Alveolar Proteinosis	Alveolarproteinose	DP-AP	DPLD_024
Bronchiectasis	Bronchiektasen	DP-BR	DPLD_028
Endstage lung disease / Lungenerkrankungen im Endstadium		ELD	ELD_999
BOS/ReTx	BOS/ReTx	ELD-B	
Restrictive	Restriktiv	ELD-B1	ELD_001
Obstructive	Obstruktiv	ELD-B2	ELD_002
Graft vs. Host Disease (GvHD)	GvHD	ELD-G	ELD_003
Pulmonary Hypertension / Pulmonale Hypertonie		PH	PH_999
Cat.1-Pulmonary Arterial Hypert. (PAH)	Kat. 1 - PAH	PH-1	PH_001
Cat. 2-PH due to left heart disease	Kat. 2 - PH bei Linksherzinsuffizie	PH-2	PH_002
Cat. 3-PH due to chronic obstructive lung disease	Kat. 3 - PH bei chronisch obstrukt	PH-3	PH_003
Cat. 4-CTEPH (chronic thromboembolic pulmonary	Kat. 4 - chronisch thromboembolis	PH-4	PH_004
Cat. 5-PH with unclear multifactorial mechanism	Kat. 5 - PH mit unklarer Genese,	PH-5	PH_005
Bronchopulmonary Dysplasia, BPD / Bronchopulmonale Dysplasie, BPD		BPD	
Neonatal Chronic Lung disease	Neonatale chronische Lungenerkr	BPD-N	
Preterm	Frühgeborene	BPD-N1	BPD_01
Fullterm	Neugeborene	BPD-N2	BPD_02
Pneumonia Infection / Pneumonie		PN	PNEU_999
CAP (Community Acquired Pneumonia)	CAP	PN-CAP	PNEU_001
viral	viral	PN-CAP1	PNEU_004
bacterial	bakteriell	PN-CAP2	PNEU_006
mixed form (viral/bacterial)	Mischform(viral/bakteriell)	PN-CAP3	PNEU_007
HAP (Hospital Acquired/Nosocomial Pneumonia)	HAP (Nosokomiale Pneumonie)	PN-HAP	PNEU_002
viral	viral	PN-HAP1	PNEU_008
bacterial	bakteriell	PN-HAP2	PNEU_009
mixed form (viral/bacterial)	Mischform(viral/bakteriell)	PN-HAP3	PNEU_010
VAP (Ventilator-assisted Pneumonia)	VAP (beatmungsassoziierte Pneu	PN-VAP	PNEU_003
Immune compromised	Immunkompromittiert durch	PN-I	
fungal (Aspergillosis, PCP, Candida)	Pilze (Aspergillose, PCP, Car	PN-I1	PNEU_011
viral (CMV, HSV)	Viren (CMV, HSV)	PN-I2	PNEU_012

bacterial	Bakterien	PN-I3	PNEU_013
Acute Lung Injury, ALI/ARDS		ALI	ALI_999
Direct Noxe	Direkte Noxe	ALI-D	ALI_001
Indirect Noxe	Indirekte Noxe	ALI-I	ALI_002
Tuberculosis / Tuberkulose		TB	TB_999
Mycobacterium tuberculosis (MTB)	Mykobakterielle Infektion	TB-MTB	
Open Tuberculosis	Offene Lungen-TB	TB-MTBO	TB_001
Pulmonary Tuberculosis	Pulmonale TB	TB-MTBP	
Extrapulmonary Tuberculosis	Extrapulmonale TB	TB-MTBE	
Drug resistant forms	AM-resist. Formen	TB-MTBR	
MDR-TB	MDR-TB	TB-MTBRM	
XDR-TB	XDR-TB	TB-MTBRX	
Non tuberculous Mycobacteria (NTM)	Nicht-mykobakterielle Infektion (N	TB-NTM	TB_006
IGRA+, no active TB (LTBI)	IGRA+, keine aktive TB (LTBI)	TB-LTBI	TB_005
Benign lesions / Gutartige Veränderungen/Erkrankungen		BR	
Hamartochondroma	Hamartochondrom	BR-1	
Lipoma	Lipom	BR-2	
Necrosis	Nekrose	BR-3	
Pulmonary infarction	Lungeninfarkt	BR-4	
Pneumothorax	Pneumothorax	BR-5	
Pleurisy	Pleuritis	BR-6	
Lung Cancer Histology/ Lungenkrebs Histologie (ICD10:C34)		**ICD-O-3**	**LC_999**
Adenoma and Adenocacinoma	Adenome und Adenokarzinome	814-849	
Acinar adenocarcinoma	Azinäres Adenocarzinom	8550/3	
Adenocarcinoma n.o.s.	Adenokarzinom o.n.A.	8140/3	
Adenocarcinoma in situ n.o.s.	Adenocarcinoma in situ o.n.A	8140/2	
Adenocarcinoma with mixed subtypes	Adenokarzinom mit gemischte	8255/3	
Bronchiolo-alveolar carcinoma, mixed mucinous ar	Gemischtes muzinöses und n	8254/3	
Bronchiolo-alveolar carcinoma, mucinous	Muzinöses bronchiolo-alveolä	8253/3	
Bronchiolo-alveolar carcinoma	Bronchiolo-alveoläres Karzinc	8250/3	
Bronchio-alveolar carcinoma, non-mucinous	Nichtmuzinöses bronchiolo-al	8252/3	
Clear cell adenocarcinoma n.o.s.	Klarzelliges Adenokarzinom o	8310/3	
Fetal adenocarcinoma	Fetales Adenokarzinom	8333/3	
Micropapillary adenocarcinoma n.o.s.	Mikropapilläres Adenokarzino	8265/3	
Mucinous adenocarcinoma	Muzinöses Adenokarzinom	8480/3	
Papillary adenocarcinoma n.o.s.	Papilläres Adenokarzinom o.r	8260/3	
Signet ring cell carcinoma	Siegelringzellkarzinom	8490/3	
Solid adenocarcinoma n.o.s.	Solides Adenokarzinom o.n.A	8230/3	
Neuroendocrine	Neuroendokrine	804-824	
Atypical carcinoid tumor	Atypischer Karzinoidtumor	8249/3	
Combined small cell carcinoma	Kombiniertes kleinzelliges Ka	8045/3	
Large cell neuroendocrine carcinoma	Großzelliges neuroendokrines	8013/3	
Neuroendocrine Carcinoma n.o.s.	Neuroendokrines Karzinom o	8246/3	
Small cell carcinoma n.o.s.	Kleinzelliges Karzinom o.n.A.	8041/3	
Typical carcinoid	Typisches Karzinoid	8240/3	
Other (rare)	Andere (seltene)	802-898	

Adenoid cystic carcinoma	Adenoid-zystisches Karzinom	8200/3	
Adenosquamous carcinoma	Adenosquamöses Karzinom	8560/3	
Carcinosarcoma n.o.s.	Karzinosarkom o.n.A.	8980/3	
Epithelial-myoepithelial carcinoma	Epithelial-myoepitheliales Kar	8562/3	
Giant cell carcinoma	RiesenzellKarzinom	8031/3	
Large cell carcinoma n.o.s.	Großzelliges Karzinom o.n.A	8012/3	
Lymphoepithelial carcinoma	Lymphoepitheliales Karzinom	8082/3	
Mucoepidermoid carcinoma	Mukoepidermoid Karzinom	8430/3	
Non-small cell carcinoma	Nichtkleinzelliges Karzinom	8046/3	
Pleomorphic carcinoma	Pleomorphes Karzinom	8022/3	
Pulmonary blastoma	Lungenblastom	8972/3	
Sarcomatoid carcinoma	Sarkomatoides Karzinom	8033/3	
Spindle cell carcinoma n.o.s.	Spindelzellkarzinom o.n.A.	8032/3	
Squamous Cell Neoplasia	Plattenepithelneoplasien	805-808	
Basaloid carcinoma	Basaloidkarzinom	8123/3	
Basaloid squamous cell carcinoma	Basaloides Plattenepithelkarz	8083/3	
Papillary squamous cell carcinoma	Papilläres Plattenepithelkarzi	8052/3	
Squamous cell carcinoma n.o.s.	Plattenepithelkarzinom o.n.A.	8070/3	
Squamous cell carcinoma in situ n.o.s.	Plattenepithelkarzinom in situ	8070/2	
Squamous cell carcinoma, clear cell type	Klarzelliges Plattenepithelkar	8084/3	
Squamous cell carcinoma, keratinizing	Verhornendes Plattenepithelk	8071/3	
Squamous cell carcinoma, nonkeratinizing n.o.s.	Nichtverhornendes Plattenepi	8072/3	
Squamous cell carcinoma, small cell, nonkeratizing	Kleinzelliges nichtverhornend	8073/3	
Mesothelial Neoplasia	Mesotheliale Neoplasien	905-905	
Mesothelioma biphasic	Mesotheliom biphasisch	9053/3	
Mesothelioma epitheloid	Mesotheliom epitheloid	9052/3	
Mesothelioma sarcomatoid	Mesotheliom sarkomatoid	9051/3	
Epithelial Neoplasia of the Thymus	Epitheliale Neoplasien des Thymu	858-858	
Thymic carcinoma n.o.s.	Thymuskarzinom o.n.A.	8586/3	
Thymoma n.o.s.	Thymom o.n.A.	8580/1	
Pulmonary metastases	Lungenmetastasen	MT	
Primary tumor colon + pulmonary metastases	Primärtumor Kolon + Lungenı	MT-1	
Primary tumor kidney + pulmonary metastases	Primärtumor Niere + Lungenr	MT-2	
Primary tumor breast + pulmonary metastases	Primärtumor Brust + Lungenn	MT-3	

Healthy Control / Gesunde Kontrollen		K	
other disease (without pulmonary disease)	krank aber nicht Lungenkrank	K-1	
Healthy (Adult)	gesund (Adult)	K-2	CON_01
Preterm	Frühgeborene	K-3	BPD_03
Fullterm	Neugeborene	K-4	BPD_04

12. Publikationsverzeichnis

2012 **Stöhr M**, Majeed RW, Edeler B, Röhrig R. Semantische Interoperabilität zwischen Rettungsdienst und Klinik. In: 1. Symposium ICT in der Notfallmedizin. Rauischholzhausen, 12.-13.06.2012. Düsseldorf: German Medical Science GMS Publishing House; 2012. Doc12notit13. DOI 10.3205/12notit13

2012 Majeed RW, **Stöhr MR**, Röhrig R: Proactive Authenticated Notifications for Health Practitioners: two way Human Computer Interaction Through Phone. Studies in health technology and informatics 180, 388

2013 Majeed RW, **Stöhr MR**, Röhrig R: Architecture of a prehospital emergency patient care report system (PEPRS). Stud Health Technol Inform. 2013;192:1151

2014 Edeler B, Majeed RW, Ahlbrandt J, **Stöhr MR**, Stommel F, Brenck F, Thun S, Röhrig R: LOINC in prehospital emergency medicine in Germany - experience of the DIRK-project. Methods Inf Med. 2014;53(2):87-91. doi: 10.3414/ME12-02-0015

2014 Majeed RW, **Stöhr MR**, Brenner T, Röhrig R. ChronoQuery: Visual Modelling of Temporal Queries for Real-Time Decision Support. Stud Health Technol Inform. 2014;205:93

2017 Mate S, Kadioglu D, Majeed RW, **Stöhr MR**, Folz M, Vormstein P, Storf H, Brucker DP, Keune D, Zerbe N, Hummel M. Proof-of-Concept Integration of Heterogeneous Biobank IT Infrastructures into a Hybrid Biobanking Network. Stud Health Technol Inform. 2017;243:100

2017 **Stöhr MR**, Helm G, Majeed RW, Günther A. CoMetaR: A Collaborative Metadata Repository for Biomedical Research Networks. Stud Health Technol Inform. 2017;245;1337

2017 Majeed RW, Xu T, **Stöhr MR**, Röhrig R. Li2b2-Façade: Simulation of i2b2 Data Warehouse Server and Client for Interaction with Other Systems. Stud Health Technol Inform 245:1275

2017 Majeed RW, **Stöhr MR**, Thiemann VS, Röhrig R, Günther A. Asynchronous Query Distribution Between Multiple i2b2 Research Data Warehouses: Li2b2-SHRINE. Stud Health Technol Inform 245:1276

2018 **Stöhr MR**, Majeed RW, Günther A. Using RDF and Git to Realize a Collaborative Metadata Repository. Studies in health technology and informatics. 2018;247:556-60

2018 **Stöhr MR**, Majeed RW, Günther A. Metadata Import from RDF to i2b2. Stud Health Technol Inform. 2018;253:40-44

2018 Majeed RW, **Stöhr MR**, Ruppert C, Günther A. Data Discovery for Integration of Heterogeneous Medical Datasets in the German Center for Lung Research (DZL). Stud Health Technol Inform. 2018;253:65-69

2019 **Stöhr MR**, Günther A, Majeed RW. Verifying Data Integration Configurations for Semantical Correctness and Completeness. Stud Health Technol Inform. 2019 Sep 3;267:66-73

2019 **Stöhr MR**, Günther A, Majeed RW. Provenance for Biomedical Ontologies with RDF and Git. Stud Health Technol Inform. 2019 Sep 3;267:230-237. doi:10.3233/SHTI190832

2020 Fischer P, **Stöhr MR**, Gall H, Michel-Backofen A, Majeed RW. Data Integration into OMOP CDM for Heterogeneous Clinical Data Collections via HL7 FHIR Bundles and XSLT. Stud Health Technol Inform. 2020 Jun 16;270:138-142

2020 Krauss E, El-Guelai M, Pons-Kuehnemann J, Dartsch RC, Tello S, Korfei M, Mahavadi P, Breithecker A, Fink L, **Stoehr M**, Majeed RW, Seeger W, Crestani B, Guenther A. Clinical and Functional Characteristics of Patients with Unclassifiable Interstitial Lung Disease (uILD): Long-Term Follow-Up Data from European IPF Registry (eurIPFreg). J Clin Med. 2020 Aug 3;9(8):2499

2021 **Stöhr MR**, Günther A, Majeed RW. ISO 21526 Conform Metadata Editor for FAIR Unicode SKOS Thesauri. Stud Health Technol Inform. 2021 May 24;278:94-100

2021 Majeed RW, **Stöhr MR**, Günther A. HIStream-Import: A Generic ETL Framework for Processing Arbitrary Patient Data Collections or Hospital Information Systems into HL7 FHIR Bundles. Stud Health Technol Inform. 2021 May 24;278:75-79

2021 **Stöhr MR**, Günther A, Majeed RW. The Collaborative Metadata Repository (CoMetaR) Web App: Quantitative and Qualitative Usability Evaluation. JMIR Med Inform 2021;9(11): e30308. doi: 10.2196/30308

2021 Ulrich H, Kock-Schoppenhauer A, Deppenwiese N, Gött R, Kern J, Lablans M, Majeed RW, **Stöhr MR**, Stausberg J, Varghese J, Dugas M, Ingenerf J. Understanding the Nature of Metadata - A Systematic Review. Journal of Medical Internet Research. 14/10/2021:25440 (forthcoming/in press)

13. Ehrenwörtliche Erklärung

„Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne unzulässige Hilfe oder Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder nichtveröffentlichten Schriften entnommen sind, und alle Angaben, die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Dissertation erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der „Satzung der Justus-Liebig-Universität Gießen zur Sicherung guter wissenschaftlicher Praxis“ niedergelegt sind, eingehalten. Ich versichere, dass Dritte von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen, und dass die vorgelegte Arbeit weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde zum Zweck einer Promotion oder eines anderen Prüfungsverfahrens vorgelegt wurde. Alles aus anderen Quellen und von anderen Personen übernommene Material, das in der Arbeit verwendet wurde oder auf das direkt Bezug genommen wird, wurde als solches kenntlich gemacht. Insbesondere wurden alle Personen genannt, die direkt an der Entstehung der vorliegenden Arbeit beteiligt waren. Mit der Überprüfung meiner Arbeit durch eine Plagiatserkennungssoftware bzw. ein internetbasiertes Softwareprogramm erkläre ich mich einverstanden.“

Berlin, 07.06.2023 _____

Ort/Datum Unterschrift

14. Danksagung

Ich danke meinem Doktorvater Prof. Dr. Andreas Günther für das mir entgegengebrachte Vertrauen und die mir gewährten wissenschaftlichen Freiheiten während meiner Tätigkeit in seiner Arbeitsgruppe. Ich danke ihm und Prof. Dr. Henning Schneider für die Zeit der Betreuung, für ihre konstruktive Kritik und dafür, dass die beiden an meine Arbeit und an meine Fähigkeiten geglaubt haben.

Ich danke Prof. Dr. Rainer Röhrig, der mein Potenzial früh erkannte und mich lehrte, dass es in der Wissenschaft um mehr geht als nur die Schaffung von Wissen. Forschung lohnt sich!

Insbesondere danke ich meinem Betreuer, Kollegen und Freund Raphael Majeed. Sein Pragmatismus hält meinem Trotz stand, durch seinen Humor bleiben Konversationen unbeschwert und durch seine Anleitung formte sich mein wissenschaftlicher Charakter.

Ich danke meinen Eltern Christine und Peter, die mir immer ein Gefühl des Willkommenseins geben. Ich danke meiner Oma Inge, meiner Oma Marga und meinem Onkel Uli, die mir auch stets ein Zuhause in ihrer Stube und in Südtirol boten.

Ich danke dem exklusiven Teil der Lehramt-Mentorengruppe und insbesondere Steffi und Jonas, die mir beide neue Perspektiven vermittelt haben und so erlaubten persönlich zu wachsen.

Ich danke den Gießen City Bitches und dem Pommes-Konvoi für Stunden und Tage der Ausgelassenheit, der Freude und des Miteinanders. Ich danke Thomas für das gemeinsame Beschreiten eines anspruchsvollen Weges seit dem Vorkurs. Insbesondere danke ich meiner besten Freundin Isabell, die mir das Geheimnis von Bulgur aufm Döner verraten hat.

Schlussendlich sehe ich mich unumgänglich gezwungen auch meiner Partnerin Kim besonderen Dank auszusprechen, da sie sich trotz aller Widrigkeiten vehement weigert ihre Rolle als die auf meine Wenigkeit bezogen dufteste Frau abzulegen.