UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Data Science Curriculum

Harry-Anton Talvik

Workflow for Transforming Health Records to OMOP Common Data Model

Master's Thesis (15 ECTS)

Supervisor(s): Sven Laur, PhD
Raivo Kolde, PhD
Sulev Reisberg, PhD

Tartu 2022

Workflow for Transforming Health Records to OMOP Common Data Model

Abstract:

In recent years, there has been a growing need in health informatics to use operational health data for research. In Estonia, medical data generated as part of daily work in healthcare institutions are exchanged in HL7 CDA format. The challenge is correctly converting the data from its exchange format for analytical use; this is a very time-consuming process.

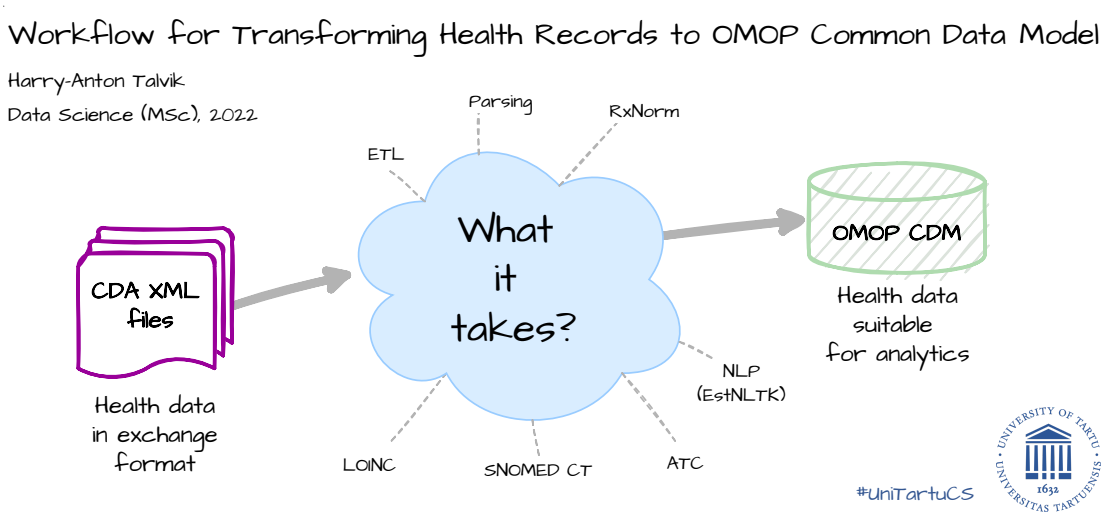
This work aims to describe the overall workflow for converting data to a common data model (OMOP CDM) for analytical purposes. As a practical solution, an improved capability for data exploration and initial information extraction from the CDA format has been created. Improved automation of the workflow environment enables rapid deployment of the solution. The resulting data in OMOP CDM is suitable for generating evidence that promotes better health decisions and better care.

Keywords:

data pipeline, OMOP common data model, clinical document architecture, observational medical data, discharge summaries, referrals, responses to referrals

CERCS: B110 - Bioinformatics, medical informatics, biomathematics, biometrics

Visual Abstract:



Töövoog tervisedokumentide teisendamiseks OMOP CDM kujule

Lühikokkuvõte:

Viimastel aastatel on tervishoiuinformaatika valdkonnas üha rohkem tuntav vajadus kasutada operatiivtöö käigus kogunenud terviseandmeid teisestel eesärkidel, teadusuuringuteks. Eestis vahetatakse tervishoiuasutustes igapäevase töö käigus tekkivat meditsiinilist teavet kliiniliste dokumentide formaadis (HL7 CDA). Väljakutse on andmete korrektne teisendamine nende transpordikujult analüütiliseks kasutuseks, see on väga aeganõudev protsess.

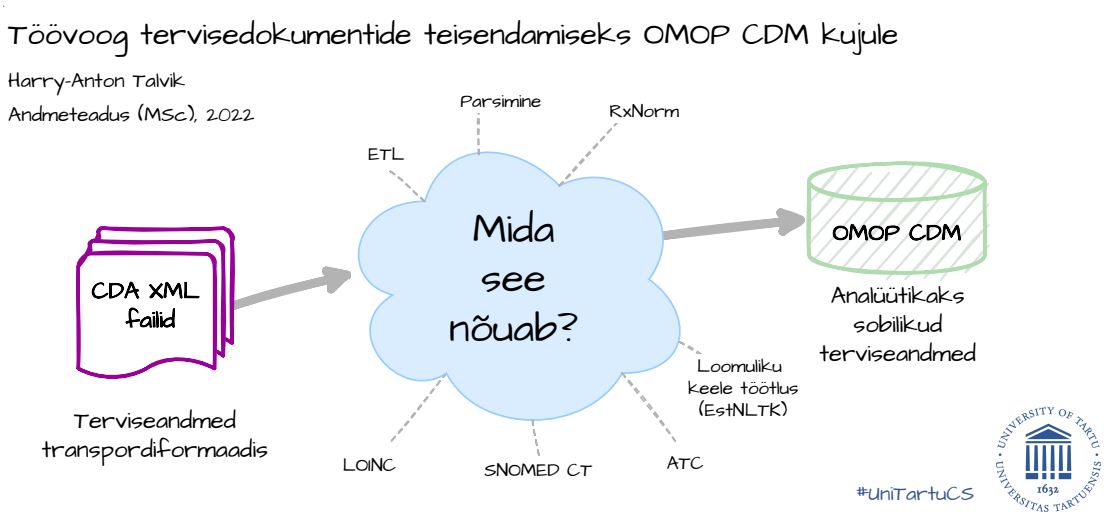
Magistritöö eesmärk on kirjeldada ära üldine töövoog andmete teisendamiseks analüütilist kasutust võimaldavale ühtse andmemudeli (OMOP CDM) kujule. Praktilise lahendusena loodi parem võimekus CDA formaadis andmetest ülevaate saamiseks ja algseks info eraldamiseks. Töövoo keskkonna täiendav automatiseerimine võimaldab loodud lahenduse kiiret kasutuselevõttu. Keskkonna kasutuse tulemusena tekkivad andmed võimaldavad omakorda tervisealase tõenduspõhise info genereerimist ja seeläbi paremate tervishoiualaste otsuste tegemist ning paremat, tõenduspõhist ravi.

Võtmesõnad:

CDA, ühtne andmemudel, OMOP, meditsiinilised vaatlusandmed, ambulatoorne epikriis, statsionaarne epikriis, saatekirjad, saatekirjade vastused

CERCS: B110 - Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika

Visuaalne kokkuvõte:



Acknowledgements

I would like to express my gratitude to my supervisors Sven, Sulev, and Raivo for guidance and positive encouragement, and UTHPC people for technical advice and support.

I also would like to express my sincere appreciation to my second half, Maarit, who is magnificent in every way.

Contents

1 Introduction

Over the past years, the field of health informatics has known intensifying need for rapid utilization of secondary data for research purposes – all stemming from the aspiration of and the need for clinical decisions to be supported by accurate, timely, and up-to-date, evidence-based clinical information.

In Estonia, information collected from clinics during actual medical practice is, to the extent, exchanged via documents in a Health Level 7 (HL7) Clinical Document Architecture (CDA) format. Standardized health information exchange is paramount for operational activities and is a good start for secondary use of data.

However, observational health studies need such data in large quantities, harmonized into a common data standard to ensure that research methods can be applied systematically to produce comparable and reproducible results. One such common data standard is provided by the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). OMOP CDM enables systematic, standardized and large-scale analytics that can be applied to clinical patient data.

1.1 Problem statement

While standards for operational health information exchange and analytical workloads exist and are established, the data transformation from one to another is a very time-consuming process, often not very robust, and messier than needed for predictable, streamlined activity. Unfortunately, there are no dedicated tools to perform or support the transforming of CDA data into OMOP CDM.

Additionally, due to the health data sensitivity, there is a need for secure, on-premises infrastructure with all the necessary tools and utilities correctly configured for initial transformation work and observational studies performed on top of the transformed data.

Ideally, we would need a tool that could consume millions of CDA documents and output an OMOP CDM database ready for further analytics, running on our own, secure infrastructure. Since there is no such thing, we have started building it ourselves.

1.2 Contributions

This thesis aims to provide a standardized environment and an architecture for transforming Estonian CDA medical records into OMOP CDM format suitable for generating the

evidence that promotes better health decisions and better care.

Over the years, various larger and smaller parts, bits, and pieces playing a role in forming the current state of the environment were contributed by many people, and the development of those was supported by several organizations directly or indirectly (see Acknowledgements). The author's most significant practical contributions in the scope of this thesis are:

1. creating a 2nd-generation CDA XML parser used for initial data ingestion;
2. automation of the infrastructure setup;
3. adding additional task-specific tooling and automation onto various steps within the environment.

The resulting software and tools are at the disposal of the Health Informatics team at the University of Tartu to help perform further studies and scientific infrastructure automation.

Furthermore, to the best of our knowledge, the entire health data processing pipeline from Estonian CDA documents (inpatient and outpatient discharge summaries, referrals, responses to referrals) to OMOP CDM has never been described before. As concise reminders, we present *Lessons Learned* from the various subsystems described to help those interested in planning their journey from data to reliable evidence.

1.3 Datasets Used

Developing good data products and pipelines would not be possible without real or at least truthful data—without this, fitness for the purpose would be questionable from the beginning. The following subsections describe briefly datasets that have informed the development needs of tools and pipelines developed for our standardized transformation environment.[1] Enhanced tools, in turn, have been employed to aid projects employing those datasets.

1.3.1 Dataset: RITA-MAITT

- Count of patients: 149 000.

[1]The author has not had direct access to all listed datasets. Thus, in such cases, tools developed primarily by the author have been employed successfully by others without any help from the author.

- Count of CDA documents: 4 974 000.
- CDA document types: Inpatient discharge summaries (3.4%), outpatient discharge summaries (54.2%), referrals (9.4%), responses to referrals (33%).
- Time period (years): 2012-2019.
- Anonymization: Done by TEHIK.
- Notes: 10% random sample of the Estonian population. The final report with the results was published in spring 2022 [SVR+22].

1.3.2 Dataset: EGCUT

- Count of patients: 196 000.
- Count of CDA documents: 5 174 000.
- CDA document types: Inpatient discharge summaries (6%), outpatient discharge summaries (94%).
- Time period (years): 2009-2020.
- Anonymization: Done by Estonian Genome Center of the University of Tartu (EGCUT).
- Notes: Only inpatient and outpatient discharge summaries. Several such datasets have been used to inform developments of CDA parsing and cleaning tools.

1.3.3 Dataset: HWISC

- Count of patients: 1 412 000.
- Count of CDA documents: 21 081 000.
- CDA document types: Inpatient discharge summaries (5%), outpatient discharge summaries (60%), referrals (10%), responses to referrals (25%); percents listed are based on year 2016 files.
- Time period (years): 2012-2016.

- Anonymization: Done by TEHIK.
- Notes: Anonymization was very thorough but also took a very long time. Unpacked, raw XML files take all together 617GB.

1.3.4 Dataset: Precise4Q

- Count of patients: 13400.
- Count of CDA documents: 440 000.
- CDA document types: Inpatient discharge summaries (4%), outpatient discharge summaries (96%)..
- Time period (years): 2000-2020.
- Anonymization: Done by EGCUT.
- Notes: More information about related project: https://precise4q.eu/.

1.4 Road Map

The thesis is composed as follows.

Chapter 2 (Background Knowledge and Related Work) introduces certain specific health informatics standards necessary in daily clinical practice and in the clinical research domain. Additionally, we look into the related work transforming Electronic Health Record (EHR) data onto OMOP CDM.

Chapter 3 (Data Engineering Practices) describes topics related to the workflow and pipeline management, Extract, Transform, and Load (ETL) paradigm, and tooling/library choices considered.

Chapter 4 (CDA Document Parsing) gives an overview of the initial data ingestion step, how and what kind of data from CDA documents is extracted, what kind of tools are used, and what are still open problems concerning this task.

Chapter 5 (Building Reusable Transformations) tackles problems related to the CDA data cleaning and reflects on specific details important for cleaning structured data and performing fact extraction from narratives.

Chapter 6 (Data Conversion onto OMOP Common Data Model) describes approaches used when mapping non-standard source codes onto OMOP CDM standard concepts.

Chapter 7 (Analytics Platform Deployment Automation) provides an overview of tools built on top of OMOP CDM that facilitate the design and execution of analyzes. We describe web-based ATLAS and underlying Observational Health Data Sciences and Informatics (OHDSI) WebAPI. Also, the approach used to automate the initial creation of the virtual machine to support CDA data ingestion, data cleaning, mapping transformations, and the OHDSI ecosystem/tooling is described.

Finally, chapter 8 (Discussion) shows the conclusion and discussion of the proposed standardized environment and architecture and presents future research perspectives.

2 Background Knowledge and Related Work

This chapter provides an introductory overview of certain specific health informatics standards playing crucial role in daily clinical practice as well as in clinical research domain.

Several clinical data exchange standards are in use worldwide, accompanied by many more terminology standards linked to them. Only some of these (CDA, ATC) are relevant in the scope of this work as we are primarily concerned about the standards currently used in Estonia. On the other hand, our focus is on the vocabularies supported or preferred by the common data model we are targeting (SNOMED CT, RxNorm, LOINC).

Thus, the first subchapter concentrates on the representation of clinical documents, particularly to the CDA standard. The second subchapter introduces the most prominent medical terminologies used in Estonia and elsewhere. The third subchapter describes environments and software platforms frequently used in clinical research activities; in the spotlight and under scrutiny is OMOP Common Data Model (OMOP CDM). The fourth subchapter dives into the related work transforming EHR data onto OMOP CDM, focusing on previous reports related to the CDA documents conversion.

2.1 Representation of Clinical Documents

2.1.1 Data Interchange Standards

Efforts to create a common data architecture for the interoperability of healthcare documents are by now several decades old. The Clinical Document Architecture has been in development since 1996, initially as the Kona Architecture [C+97]. Based on that, the HL7 Document Patient Record Architecture (PRA) draft for defining the semantics and structural constraints necessary for the exchange of clinical documents was proposed around 1997. The intention of the PRA was to be a common data architecture that can accommodate a diverse set of records and requirements [DAB+99].

Further developments led this work to become known as HL7 Clinical Document Architecture Framework, Release 1.0, an American National Standards Institute (ANSI)-approved HL7 Standard [DAB+01]. Additional developments, particularly in the semantic representation of clinical events, delivered a new major version of the standard, CDA, Release Two (CDA R2) [DAB+06]. This version became an ANSI-approved HL7 Standard in May 2005, and is the version used in Estonia since 2008.

2.1.2 Clinical Document Architecture Standard

The HL7 CDA is a document markup standard for the structure and semantics of an exchanged clinical document. CDA documents are encoded in Extensible Markup Language (XML), W3C XSD (XML Schema Definitions) language[2] is used to define the elements and attributes of a CDA document. Markup elements get their meaning from HL7 Reference Information Model (RIM), a domain-independent ontology capturing things that are true about the whole world [Elk12]. Together with HL7 Version 3 data types, there is an ability to incorporate concepts from standard coding systems such as Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) or Logical Observation Identifiers Names and Codes (LOINC).

Figure 1. Overall structure of a CDA R2 XML document

A CDA document is comprised of two main parts (Figure 1). The document header sets the context for the clinical document – it describes demographic information about the patient along with other information about the document itself, like who created it and when the document was written. Additionally, more technical information like document type and corresponding standard version is captured in the header.

[2] See https://www.w3.org/standards/xml/schema

The document's body is divided into sections containing the human-readable narrative text, which can contain more structural forms like lists or tables. Sections might also include machine-processable information called *entries*. The intention here is that entries contain the same data described in Narrative Block but in a format that a computer can consume and use directly, e.g., for decision support applications [DAB+06].

Inside entries, the HL7 Clinical Statement model is used to write machine-readable clinical statements like those that are captured via `<procedure>` ("an act whose outcome results in the physical alteration of the subject") or `<substanceAdministration>` ("an administration of a particular substance, e.g., a medication, immunization or other substance to a patient") elements [Boo11]. However, this results in rival fields for the same content, and inconsistencies might ensue. Also, when not displayed to the doctor, machine-processable fields are of less value.

Inside `<section>` element, `<text>` and `<title>` elements are meant to be rendered to provide the human readable display whereas the `<code>` element acts as an important marker in programmatic processing (see chapter 4, CDA Document Parsing for more details). The table model in CDA documents is almost identical to the Extensible HyperText Markup Language (XHTML) table model.

2.2 Terminology Standards

In the following, we cover terminologies used in Estonian data (ATC, SNOMED CT, LOINC) and the ones where we have to map this information in the context of OMOP CDM (SNOMED CT, LOINC, RxNorm).

Without knowing how ATC or RxNorm are assembled, it would be hard to create quality mappings. Likewise, the same applies to SNOMED CT and LOINC—one has to know how target systems are built to map anything on them reliably.

2.2.1 Anatomical Therapeutic Chemical Classification

Anatomical Therapeutic Chemical (ATC) classification is recommended for worldwide use to serve as a tool for drug utilization monitoring and research in order to improve quality of drug use [WHO18b].

The ATC classification system divides the drugs into different groups according to the organ or system on which they act and according to their chemical, pharmacological and therapeutic properties [ATC22, WHO18a].

The ATC classification system is a strict hierarchy with semantic identifiers [Cim98], it has five hierarchical levels:

1. **Anatomical main group** – characterised by a letter of the alphabet like `A` (ATC code) with corresponding name `ALIMENTARY TRACT AND METABOLISM`;

2. **Therapeutic subgroup** – characterised by a 2-digit number like `01` giving us 2nd level ATC code `A01` with name `STOMATOLOGICAL PREPARATIONS`;

3. **Pharmacological subgroup** – again, characterised by a letter of the alphabet giving us ATC code `A01A` (and for a given example the class name is the same as on previous level);

4. **Chemical subgroup** – characterised by a letter of the alphabet, giving us `A01AB` for `Antiinfectives and antiseptics for local oral treatment`;

5. **Chemical substance** – characterised by a 2-digit number specific to each chemical, running example giving us ATC code `A01AB09` assigned to an ingredient `miconazole`; notably, shown code is given to *an ingredient for a specific therapeutic intent* (i.e. for stomatological use in this example) and not for `miconazole` in general.

Grouping levels are often accompanied by descriptions of what the group comprises[3] or what is classified elsewhere e.g. *"Products used in common minor infections of mouth and throat are classified in R02, e.g. cetylpyridinium."*

Last, chemical substance level is often assisted by Defined Daily Dose (DDD)[4] describing *Route of administration (Adm.R)* and *Unit (U)*, e.g. for `miconazole` DDD is 0.2 grams, when used for stomatological treatment, and when administered orally.

By ATC principles, a medicinal substance can be given more than one ATC code if it can have therapeutic applications on different anatomical section [WHO18a]. As an example, we can see[5] that `miconazole` is classified as `A07AC01` (belonging into 3rd level subgroup `INTESTINAL ANTIINFECTIVES` (`A07A`)), and five more ATC codes on

[3] See `https://www.whocc.no/atc_ddd_index/?code=A01AB09` for grouping level descriptions of our example, `A01AB09` – `miconazole`

[4] General principles for DDD assignment are described at `https://www.whocc.no/ddd/definition_and_general_considera/`

[5] Search for `miconazole` via `https://www.whocc.no/atc_ddd_index/`

chemical substance level are assigned, including D01AC52 with a class name miconazole, combinations.

The last example means that drugs are often underspecified in ATC, meaning they do not specify which substances can be associated with given substance or what the DDD is for miconazole in this case. This is a concern when we want to map this information onto other terminologies, e.g. when mapping ATC to clinical drugs in RxNorm.

ATC in Estonia. The ATC implementation and maintenance in Estonia is curated by the Estonian national Agency of Medicines. They oversee also the Register of Medicinal Products, the purpose of which is to keep account of all medicinal products and their packages sold in Estonia. Currently, the registry

- keeps track of over 44 000 human medicine packages, including items like
  - package name,
  - ATC code,
  - active substance,
  - strength,
  - dosage form;
- maintains more than 4300 substance level ATC codes, including about 3800 unique substances.

Going back to our example, currently is miconazole-based stomatological treatment (A01AB09) available as DAKTARIN with dosage form *oral gel* and strength *20mg 1g*, in total 40g a piece.

The ATC classification is one of the official classifications[6] used by Estonian nation-wide Health Information System (EHIS); exchanged CDA documents carry presented information mostly in sections marked as DRUG *(Medicines issued)* or IMM *(Immunization)*, and minor amounts in DRUGSCH *(Treatment regimen)* subsections (based on document types available via RITA-MAITT dataset, see section 1.3.1).

[6]Referenced from national information system management system RIHA: https://www.riha.ee/Infos%C3%BCsteemid/Vaata/digilugu

2.2.2 Logical Observation Identifiers Names and Codes

LOINC provides a set of universal names and ID codes for identifying laboratory and clinical test results [FMD+96, MHS+03]; it was developed to serve as a definitive standard for identifying clinical information (tests, measurements, observations) in electronic reports. One can think of, for example, an observation as a question and the observation result value as an answer to it [Reg22c].

In following, we give a short introduction to the LOINC coding system based on the information freely available in its Users' Guide [Reg22b].

Anatomy of a LOINC Term Name. LOINC codes distinguish a given observation (test ordered/reported, survey question, clinical document) across six major dimensions called *LOINC Parts* aka *axes* [Reg22a]. First five parts are required and the last one, *Method*, is optional.

1. **Component (Analyte).** The substance or entity being measured or observed. Examples being:
   - a substance like `Glucose` or `Sodium`,
   - a derived measurement like `Hematocrit`,
   - `Aldosterone^1H post 25 mg captopril PO`, a steroid hormone with information about associated oral loading test performed.
2. **Property.** The specific observed characteristic or attribute of the analyte.
   - Examples: `Substance Concentration` (`SCnc`), `Catalytic Ratio` (`CRto`, a ratio of the enzymatic activities reported as %), `Volume` (`Vol`).
3. **Time Aspect.** The interval of time over which an observation was made.
   - For most of LOINC terms the time aspect is just "point in time" (`Pt`).
   - A Time Aspect other than `Pt` is often associated with a Property of "Rate".
   - Besides direct specification of a time window (e.g. `5 minutes` (`5M`), `3 months` (`3Mo`)) indirect specifications of a time window is possible (e.g. `Duration of the study` (`Stdy`), `Duration of an encounter (hospital stay, visit)` – abbreviated as `Enctr`).

- By default, a mean value over the time period in questions is assumed; if this is not the case, one can communicate this via an optional modifier, e.g., `8H^max heart rate` would be the highest heart rate observed over 8H.

4. **System (Specimen).** The specimen or thing upon which the observation was made.
   - Includes familiar specimens such as `Serum` (`Ser`), `Serum/Plasma` (`Ser/Plas`, either Serum or Plasma is a suitable specimen for measuring a particular analyte), `Leukocytes` (`WBC`).
5. **Scale.** How the observation value is quantified or expressed. Most prominent scale types are:
   - Quantitative (`Qn`) – The result of the test is a numeric value that relates to a continuous numeric scale; can be reported as an integer, ratio (`1:64`), real number (`-0.5`), or range (`1-12`).
   - Ordinal (`Ord`) – Terms have results that can be placed in rank order, e.g., `mild`, `moderate`, `severe`.
   - Nominal (`Nom`) – Nominal or categorical responses that do not have a natural ordering (e.g., categories of appearance such as, `yellow`, `clear`, `bloody`).
6. **Method.** *[Optional]* A high-level classification of how the observation was made. Methods are only needed when the technique affects the clinical interpretation of the results, i.e. only when different methods give clinically significant different results [MHS+03].

LOINC creates terms both for individual variables (reportable tests or clinical measurements) and collections of such variables. LOINC terms are created to represent two kinds of collections:

- **Panels**, where the child elements are enumerated, and
- **Documents** where the terms represent general information collections whose contents are not explicitly enumerated.

Anything not essential for uniquely identifying the test is left out of the LOINC name (e.g. pricing or cost, test interpretation, test instrument used, who did the test, where the test was performed, specific details about the specimen and how it was collected) [Vre16].

LOINC in Estonia. Estonian laws require the transmission of standardised and machine-readable laboratory test results and data by a healthcare provider (HCP) to the EHIS [Vab22b, Vab22a]. The lists used in the laboratory data service are based on either international terminology or locally generated coding:

- LOINC terminology is used for coding laboratory analyses (analysis code and name, parameter code and name).
- The preferred terminology for laboratory analysis responses and other laboratory-specific lists has become SNOMED CT. Where it is not practical to use SNOMED CT to assemble an enumerated lists of values, the local coding will be used to generate the list [EEK+21].

The lists are managed by the Estonian Society for Laboratory Medicine (ELMÜ). The creation of a new list or the introduction of changes to the lists is initiated by the ELMÜ and the requested actions are coordinated with Health and Welfare Information Systems Centre (TEHIK). Coordinated lists in the TEHIK Publication Centre[7] are also available on the ELMÜ website [Ees22].

Tooling. Mapping local codes to LOINC codes can be aided by the official Windows-only utility Regenstrief LOINC Mapping Assistant (RELMA); what to pay attention to in the process, is discussed in the section 6.1.1, Mapping Vocabularies – Why Caution is Warranted. Estonian LOINC management efforts are supported by the e-Laboratory Management Application (eLHR), through this are managed also local codes (representing about 10% of the total number of eLHR codes [Pal21]).

2.2.3 RxNorm

RxNorm is US-specific terminology providing normalized names for clinical drugs and links its names to many of the drug vocabularies commonly used in pharmacy management and drug interaction software [RxN22]. The goal of RxNorm is to allow computer systems to communicate drug-related information efficiently and unambiguously.

The creation of RxNorm was motivated by the need for a single, standard, multipurpose terminology for representing medications as the Food and Drug Administration's National Drug Codes (NDC) [NDC22] were not deemed suitable for tasks involving

[7] See https://pub.e-tervis.ee/

automated decision support, quality assurance, healthcare research, reimbursement, and mandatory reporting. Notably, NDCs did not provide consistent enough way to identify the essential components of prescriptions for purposes of facilitating electronic capture of such data in EHRs [NZK+11].

As a terminology, RxNorm is built on and derived from other terminologies. RxNorm reflects and preserves the meanings, medication names (both generic and brand), info about dosage, route of administration, ingredients, as well as attributes, and relationships from its sources. Currently, RxNorm receives drug names from 15 different sources, including ATC and US Edition of SNOMED CT. RxNorm itself is part of Unified Medical Language System (UMLS), a huge repository of biomedical vocabularies and standards integrated and managed by the US National Library of Medicine [Bod04].

In US, RxNorm names and codes are used to represent medication names in the Continuity of Care Document (CCD) [NZK+11], which is pretty much the equivalent of ATC being used in Estonian CDA documents.

2.2.4 Systematized Nomenclature of Medicine – Clinical Terms

SNOMED CT is considered to be the most comprehensive, multilingual clinical healthcare terminology in the world [BG21]. It is an interface terminology, meant to be used for clinical data entry and display in the user interfaces of healthcare IT systems; thus, SNOMED CT needs to be integrated with IT systems before use [Bha15], a challenge of its own.

SNOMED CT in Estonia. Estonian CDA documents use SNOMED CT codes, or more formally SNOMED CT Identifiers (SCTIDs) in various different ways. One prevalent use is employing them as contextual qualifiers throughout *anamnesis* and *objective findings* sections of outpatient summary documents. Structured parts (i.e. document parts under `section/entry/`) convey this information inside `observation/code/` elements, see an example in Listing 1; usage of SCTIDs is pointed out via comments in XML.

Specific allowed or suggested use of SNOMED CT codes is described via guidance made publicly available at the central publishing environment for medical classifications and standards, managed by TEHIK [PUB22]. This Publication Centre is also the place from where one should find all the classifications used in Estonian health Information exchange (HIE), and via this codes currently in active use, as well as the ones used in the

```
<!-- Example from
       ClinicalDocument/component/structuredBody/component/section
     where code/@code="OBJFIND". -->
<entry typeCode="COMP">
    <observation classCode="OBS" moodCode="EVN">
        <!-- 64049009: 'Examined for (contextual qualifier) (qualifier value)'   -->
        <code code="64049009" codeSystem="2.16.840.1.113883.6.96"
                              codeSystemName="SNOMED CT" displayName="Staatus"/>
        <statusCode code="completed"/>
        <effectiveTime value="20170422"/>
        <entryRelationship typeCode="COMP">
            <observation classCode="OBS" moodCode="EVN">
                <!-- Human-readable phrase (term) for SCTID 364048003:           -->
                <!--    in English: 'Respiratory observable (observable entity)' -->
                <!--   in Estonian: 'Hingamissüsteemiga seotud näitajad'         -->
                <code code="364048003" displayName="Hingamissüsteem"
                      codeSystem="2.16.840.1.113883.6.96"
                      codeSystemName="SNOMED CT"/>
                <text>*22.04.2017 MARI MAASIKAS: vesikulaarne HK</text>
                <interpretationCode code="FINDING" displayName="leid"
                      codeSystem="1.3.6.1.4.1.28284.6.2.1.55.2"
                      codeSystemName="Läbivaatuse valikvastused"/>
            </observation>
        </entryRelationship>
    </observation>
</entry>
```

Listing 1. SNOMED CT codes in Estonian CDA document, objective findings section.

past.

SNOMED CT Estonian Extension. TEHIK is also the national release center for SNOMED CT and responsible for authoring the Estonian extension [AKL22]. The Estonian extension contains:

- Value sets used in the data exchange with national health information system (distributed as SNOMED CT reference sets);
- Nationally added content which is specific to Estonian use cases;
- Estonian translations to all concepts that are used in any of the national or cross-border value sets;
- Estonian translations to the core set of concepts (continuously expanding) [Int22].

Expressed in concrete figures, this means that SNOMED CT Estonian extension contains 28 refsets (including ca 10 000 concepts), about 12 000 translations and about 450 local concepts [Lin21b].

Tooling. Sheer size and complexity of SNOMED CT warrants tools for its management, and SNOMED CT Estonian Extension is here no exception. To aid with pre-release check of extension's release files a functional prototype of the review tool has been built [Lin21a].

2.3 OMOP Common Data Model

As of 2022, it is expected that clinical decisions will be supported by accurate, timely, and up-to-date clinical information. It is expected that clinical decisions reflect the best available evidence. However, conducting research across disparate observational databases, both in a centralized environment and in a distributed research network, has been hindered by technical challenges.

The OMOP CDM has been designed as an answer to those challenges—it is a mechanism to standardize the structure, content, and semantics of observational data and to make it possible to write statistical analysis code once that could be reused at every data site [OHDSI20, ORR+12]. With the help of OMOP CDM analytics can be remote and run behind firewalls. Importantly, to be suitable for the purpose, the CDM aims to provide data organized in a way optimal for analysis rather than to address the operational needs of health care providers or payers [OHDSI20].

The key here is the employed approach, a strict standardization.

OMOP CDM and OMOP Standardized Vocabularies standardize everything about needed data:

- **data structure** constrains used tables, fields, and data types;
- **data content** is controlled by vocabularies used to codify clinical domains like conditions, drugs, measurements taken or procedures performed (see Section 6.1.2 for more information);
- **data semantics** sets forth conventions about used meaning.

Among other design principles, technology neutrality and scalability are design elements the development of the CDM follows; those are not afterthoughts. The CDM does not require a specific technology—any popular relational database like SQL Server or PostgreSQL is supported. Support for data warehouses like Amazon Redshift hints that the CDM is optimized for data processing and computational analysis on databases with hundreds of millions of persons and billions of clinical observations.

2.4 Related Work

The adoption of the OMOP CDM has been in the midst of a perfect storm in Europe and worldwide for several years. There is sizable interest from data custodians, academia, the pharmaceutical industry, patient organizations, and regulators to collaborate on implementing the OMOP CDM and federated data network in Europe [EHD22].

Justifiably, there are many reports and feasibility studies about various conversions from local source data into the CDM format [MLS+18, Lan20, LDD+19, LDD+20, ZMB+13, SJD+21, CST+20, LAAB+21, HRSG19, BOD+21, FSG+20, TTC+20].

However, we managed to find only one report about converting CDA documents into OMOP CDM [JKY+20].

Here, researchers from Seoul National University Bundang Hospital analyzed and converted about 20 000 referral CDA documents accumulated for over ten years in a tertiary general hospital in South Korea. The general process involved CDA parsing, data cleaning, standard vocabulary mapping, CDA-to-CDM mapping, and CDM conversion. The quality of CDM data was then evaluated using the Achilles Heel and visualized with the Achilles tool [DRS+22]. Parsing and cleaning steps employed regular expressions; otherwise, technical details provided are scarce.

3 Data Engineering Practices

Data engineering and data mapping are in academic circles usually regarded as a grunt-work not worthy of authorship nor acknowledgements. However, it has been pointed out that comprehensively and transparently reported methods of Electronic Health Record (EHR) data extraction and transformation are at least as important as subsequent statistical analysis and interpretation [KAA+21].

In the following, we describe central concepts behind technologies used to build reusable transformations—ETL and workflows. We conclude with a discussion about the choice of the workflow management tool used to build out our transformations and other possible options available today.

3.1 Extract, Transform, and Load

In general, ETL is a process that restructures the data from one place to another. It has been recognized that the ETL process can be particularly burdensome when one has to support multiple, large data sources and update them regularly [BBS+16]. In our case at hand, to get from the native/raw CDA data to the OMOP CDM, we have to create and use an ETL process. Optionally, we can use several smaller ETL processes throughout separate phases.

3.2 Workflow Management

The need to perform long-running ETL processes repeatedly and reliably asks for some kind of means to govern the complexity arising from interconnected extraction/transformation/loading tasks—they have to be run in the right order, with clear success/failure reports and options to run failed tasks again.

The framework to govern such complexity is known as *Workflow Management*, and it is in most part carried out via selected workflow engine, a tool that helps to manage data pipelines that conduct ETL jobs and tasks within them.

3.3 Workflow Management Tools

A selected workflow engine must also support a *T* in ETL, i.e., pipelines of tasks that transform and model data for a new purpose. However, data transformation is often

context-specific and can be written in a language familiar to data engineers and data analysts, such as SQL or Python [Den21]. Thus, to support analysts, the workflow engine of choice should employ the favored language as the primary option or provide simple means to connect tasks written in preferred languages.

There are various workflow creation frameworks and libraries available. Over time, the popularity of some of them fades and is replaced by new options that are more competitive and offer better usability or features that the market has been craving.

The choice for the workflow engine currently employed in our environment was made already around 2015-2016. More specifically, we use Luigi [LUI18] for structural data cleaning and fact extraction (Section 5). At the time, Luigi won by providing the following:

- central scheduler to queue competing jobs;
- graphical overview of processes in various states (success/failure/in progress) via web interface;
- use of Python as a primary language.

Other contenders at the time were Drake,[8] Scientific Luigi,[9] Crunch, [10] Dagobah,[11] and Airflow.[12] Modern contenders to weight today are for example Prefect,[13] Dagster,[14] and Ploomber.[15]

3.4 Lessons Learned

- In general, Luigi has served us well. Building, connecting, and running tasks even for testing has been relatively easy as Luigi offers a local scheduler and is just a library in a Python environment.

[8] See https://github.com/Factual/drake
[9] See https://github.com/pharmbio/sciluigi
[10] See https://dev.arvados.org/projects/arvados/wiki/Crunch
[11] See https://github.com/thieman/dagobah
[12] See https://github.com/apache/airflow
[13] See https://github.com/PrefectHQ/prefect
[14] See https://github.com/dagster-io/dagster
[15] See https://github.com/ploomber/ploomber

- Troublesome has been getting an overview of the reasons why some long-running processes are stalled. This has more to do with our approach to instructing tasks (not enough granular supply of input data) and less with Luigi itself.

4 CDA Document Parsing

> However, as every parent of a small child knows, converting a large object into small fragments is considerably easier than the reverse process.
>
> *Andrew S. Tanenbaum, Computer Networks, 4th ed., p. 428.*

This chapter describes our current paths taken to locate and extract parts from CDA documents, a first step enabling downstream jobs to pinpoint valuable signals about the topic we care about—health. While breaking a large tree-like document into fragments is conceptually straightforward, gaining insight into most value-rich sub-parts takes some wrangling and building a network of waypoints.

We open up the theme, first discussing the merits of different formats (4.1, Exchange *versus* Storage), followed by the problem statement in the realm of value extraction coverage (4.2, Problem Statement). The third subchapter (4.3, Architecture) describes our current implementation of the CDA XML Parser and its evolution. The last subchapter (4.4, Tools and Validation) tackles challenges around the explorability of the thick forest of CDA documents. It gives an overview of tools developed to mark down the grid of control points to guide us. Also, additional options to gain even more granular insight on the CDA document section level are prototyped and discussed.

4.1 Exchange *versus* Storage

The HL7 CDA standard promotes the longevity of clinical records [DAB+01]. A single CDA document from yesterday or five years ago can be rendered in a web browser using appropriate Extensible Stylesheet Language Transformations (XSLT) stylesheets. Using information systems based on such a mechanism, a general practitioner (GP) can retrieve previous information or learn new facts about his patient, for example, the latest results of ordered analyses. So, in this case, it is enough to use the document's original form, its interchange format.

Other uses (e.g., visualizing a patient's health history over the years, large-scale health analytics) require more complex approaches where stored information could be

queried across documents and processed more comprehensively. For example, even simple visualizations would require consistent approaches for representing values and units. However, the CDA standard is scoped by the exchange, independent of transfer, storage, or further processing.

4.2 Problem Statement

In essence, as implied in the previous section (4.1, Exchange *versus* Storage), in many cases, we need storage that enables more comfortable and more powerful ways for information retrieval and transformations. Once we have chosen a storage system, we need to extract the data from the exchange format and store it in the system in such a way that the information is not lost. At the same time, we should be able to fully leverage all the features we thought we would need when we chose the system.

The extraction of information from CDA documents requires XML parsing, which can be performed to varying degrees.

One extreme would be to target all the XML nodes (elements, attributes, text nodes), i.e., essentially build the same document tree in another format (e.g., in JavaScript Object Notation (JSON)). A tree in another format might be the solution for specific situations (e.g., when the tree in a new format can be queried somehow more comfortably or processed more efficiently).

Another path would be to extract only deliberately chosen values with enough context information not to lose the anticipated utility. An example would be:

- capture International Classification of Diseases (ICD) diagnosis codes;
- when ICD-10 and ICD-11 are both in use, then capture the coding system information as well (or better, one could do it by principle, always);
- keep track of whether an assigned diagnosis is a primary or concomitant disease;
- keep track of statistical diagnosis type (first or repeat occurrence);
- keep track of the date when the diagnosis was assigned.

We can go on like this, and eventually, we will be done—we will have covered all the fields of interest. However, the open problem is—precisely how do we know that we have covered all the possible sub-branches that may occur in certain types of CDA documents?

Or, how much and what kind of information from which part of the document is not yet covered? *"Are we there yet?"*

Additional angle for the same theme would be the following:

1. Let us say that a parser covers all of the interest area.
2. With years passing, CDA template standards are developed further (e.g., more structured parts added into relevant sections).
3. New versions of the template standards are released and, after some time, also in effect (i.e., are first suggested and then required).

How do we know how much the same parser now captures our area of interest? How could we correctly pinpoint the [structural] data drift?

We do not have solutions yet, but we are scratching the surface by planting lighthouses showing us the way. The following section will briefly cover our current CDA XML Parser architecture. After that, we will look into navigational aids helping us sail the stormy sea of CDA document's content—tools that help us find out the content we want to introduce to the parser.

4.3 Architecture

We have implemented a CDA XML Parser that pulls selected information out of CDA documents and stores it in a relational database for further processing. In fact, current implementation is a 2nd incarnation of this tool, while the first one was born around 2014.

4.3.1 Early History

> Everything should be made as simple as possible, but no simpler.
>
> *Compressed version of lines from a 1933 lecture by Albert Einstein*

Initially, the parser was written in Perl, used `XML::Simple`[16] as its core library, and pushed extracted content into a MySQL database via `DBI`,[17] Perl's standard database

[16] See https://metacpan.org/pod/XML::Simple

[17] See https://dbi.perl.org/

interface module. The parser handled 16 distinct top-level sections from two types of CDA documents—inpatient[18] and outpatient discharge summaries.[19]

For initial needs, the solution was pulling its weight more than enough. Over time, needs grew, and the parser needed workarounds to deal with mixed content and XHTML-like tables in CDA documents—this was not supported by the underlying XML parsing library [McL18]. At the same time, ergonomics of managing exact versions of dependent libraries became increasingly tricky on newer operating systems,[20] and the popularity of Python and usability of its dependency management advanced remarkably.

4.3.2 Current Implementation

> A complex system that works is
> invariably found to have evolved from
> a simple system that worked.
>
> *Gall's law*

Today, CDA XML Parser runs on Python 3.10 with dependencies managed via an open-source, cross-platform, language-agnostic package manager Conda installed via Miniconda [BSGC19, Ana22]. Compared to the early history, the core architecture and general execution flow have not changed much. However, a move to the database with schemas and roles (PostgreSQL), tight setup validation, better ergonomics of the execution, additional post-parsing steps, and establishment of a test environment are all significant additions (Figure 2).

Technically, CDA XML Parser consists of three main parts:

1. The parser core, a Python program using `lxml`[21] [BFB+21] as its core library for parsing XML content.

2. Scripts and configuration files handling parallel execution as well as various precursory and post-parsing tasks like:

    - validating configuration of the current setup (*"Are we ready for parsing?"*);

[18] In Estonian: *Statsionaarne epikriis*
[19] In Estonian: *Ambulatoorne epikriis*
[20] The CDA XML Parser has been mostly used on Debian and CentOS flavors of Linux
[21] See `https://lxml.de/`

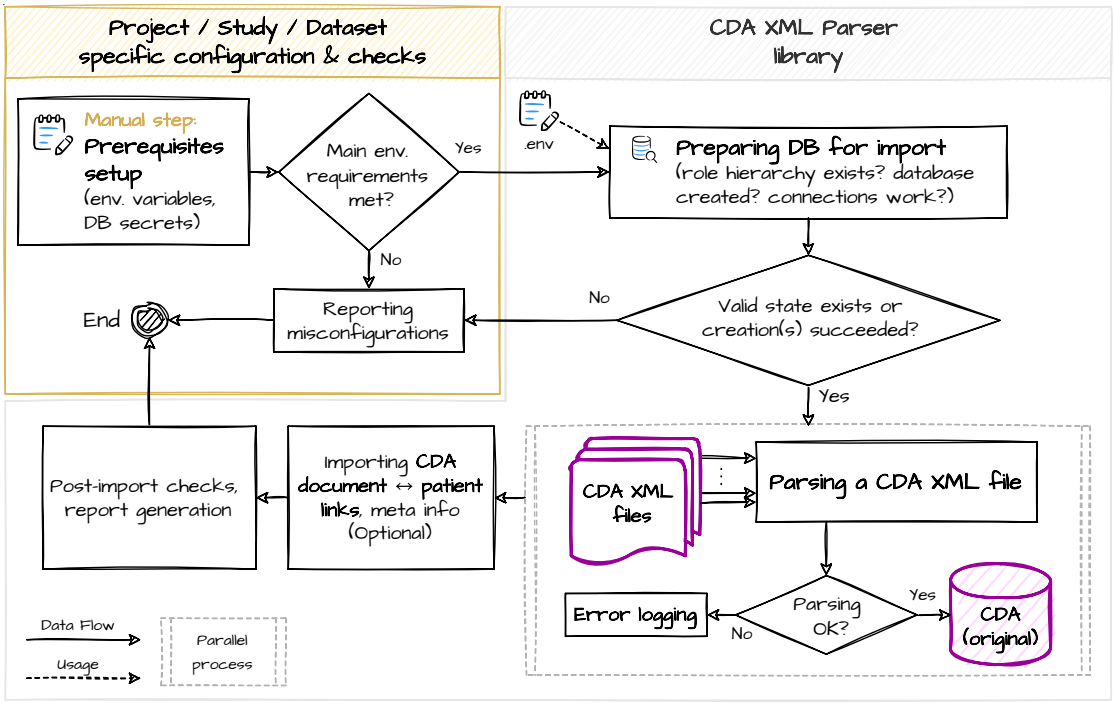


Figure 2. CDA XML Parser execution flow

- creation of the target PostgreSQL database with needed roles, schemas, tables, views, indices, and PL/pgSQL[22] functions upon a need;
- processing parsing logs (extracting error information);
- performing post-import sanity checks;
- optimizing (vacuuming) database tables.

3. Docker-based test environment with many regression tests supporting refactoring and fast development.

Recent parser rewrite from Perl to Python was greatly supported by existing end-to-end tests ensuring that what was extracted and tested before was also extracted and inserted into the database with the updated parser the same way, bit-by-bit. The updated parser generally extracts more content because it can also handle XML namespace prefixes used in some CDA documents throughout the file. Usually, namespace prefixes are used only for Estonian extensions (e.g., from the Estonian CDA document header, one

[22] See https://www.postgresql.org/docs/current/plpgsql.html

can find information about a doctor's specialty from `ext:asLicencedEntity/ext:id` element,[23] here, `ext` is a namespace prefix).

4.4 Tools and Validation

> If we have data, let's look at data. If all we have are opinions, let's go with mine.
>
> *Jim Barksdale*

Now, coming back to our previously mentioned challenge (see section 4.2, Problem Statement), we need help getting an overview of the structure of the CDA documents, so we know which areas and sub-branches to review in more detail if needed. Such insight is essential to extend the parser's ability to extract more information—guidelines/standards describing CDA document types and closer examination of samples are not enough. Likewise, having modified a parser, one wants to know the progress, e.g., how much more (or, in the negative case, less) content has been extracted. We will describe some approaches toward those goals in the following subsections.

4.4.1 Validating Parsing Progress

Improvements in parsing can be monitored by looking at the number of records in the tables and the percentage of `NULL` entries in the table fields (Listing 2).

Shown post-parsing summaries are generated automatically after parsing and are in Markdown format, suitable for diffing and storing under revision control.

[23] Full XPath is `/ClinicalDocument/author/assignedAuthor/ext:asLicencedEntity/ext:id`

```
# Summary of cda_original_20220204

* Number of imported epicrises: 5174652

## Number of parsed rows

| Table               |      Count |
|:--------------------|-----------:|
| allergy             |      88183 |
| allergy_entry       |      38718 |
...

## Fraction of null entries in tables

| Table               | Field                  | Null entries |
|:--------------------|:-----------------------|-------------:|
| allergy             | text                   |        2.18% |
| allergy_entry       | code                   |        3.93% |
| allergy_entry       | material               |       15.91% |
| allergy_entry       | original_text          |       97.30% |
...
```

Listing 2. Example of a CDA XML parsing summary (raw Markdown; partial output)

4.4.2 CDA XML Structure Probing via Object Identifiers

> They look like an IP address on steroids.
>
> *Keith W. Boone about OIDs*

The CDA standard uses Instance Identifier (II) data type that uniquely identifies a thing or object (e.g., medical record number, order id); IIs are defined based on Object Identifiers (OIDs) [SBM+04]. Mostly, the II data type is associated with `<id>`, `<setId>`, `<templateId>` and `<typeId>` elements. Those elements have fixed attributes to accommodate identity information. Epigraph's author mentions in *The CDA™ Book* [Boo11]: "Identifiers have one part that ensures uniqueness, found in the `root` XML attribute, and an optional part found in the `extension` XML attribute that can be used to represent the rest of the identifier when necessary."

To make it more clear, let us look at an example. Every CDA document's header contains an element like `<templateId root=". . . " extension=". . . "/>` with OIDs as values, e.g., `root` containing `1.3.6.1.4.1.28284.6.1.1` and `extension` containing `1.3.6.1.4.1.28284.6.1.1.64.4`. In the given case, this informs us that we are dealing with a CDA template for responses to referrals,[24] more precisely with Version 4 of it,[25] one of the standards for medical documents.[26]

Another heavy use for OIDs is identifying the coding systems employed (see Listing 1 for an example, keep an eye on `codeSystem` attributes).

We want to extract all these OIDs with related context, from all `root` and `codeSystem` attributes across all elements in all the documents we have—this gives us some sense and overview what we are dealing with. As for a context:

- we keep track of the XPath from where the attribute was found;
- in the case of `root` attribute, we also extract `extension`'s value;
- in the case of `codeSystem` attribute, we extract values of commonly accompanied attributes (`codeSystemName`, `code`, `displayName`).

Based on this logic, we built an XSLT stylesheet that extracts this information via `xsltproc`,[27] a command line XSLT processor and writes it out into a CSV file, which then can be imported into the database via `psql`,[28] a terminal-based front-end to PostgreSQL.

The resulting table allows now to inspect what coding systems we have at all, from where to find, e.g., LOINC codes, or from where we could extract ATC codes (Listing 3).

Going forward, we can query for specific coding system values to assess the feasibility of additional extraction benefits. Possibilities are not endless, but now we have a tool that helps us dig into the data set and supports making different decisions.

[24] See `https://pub.e-tervis.ee/oids.py/viewform?oid=1.3.6.1.4.1.28284.6.1.1.64` – *Saatekirja vastused*

[25] See `https://pub.e-tervis.ee/oids.py/viewform?oid=1.3.6.1.4.1.28284.6.1.1.64.4` – *Versioon 4*

[26] See `https://pub.e-tervis.ee/oids.py/viewform?oid=1.3.6.1.4.1.28284.6.1.1` – *Meditsiinidokumentide standardid*

[27] See `https://gnome.pages.gitlab.gnome.org/libxslt/xsltproc.html`

[28] See `https://www.postgresql.org/docs/current/app-psql.html`

```sql
SELECT oid_value, xpath
FROM epi_oid_info
WHERE oid_id LIKE '2.16.840.1.113883.6.73%';
-- We are showing here a possible snapshot of answers.
-- Below, '...' in 'xpath' column means that common parts like
--    '/ClinicalDocument/component/structuredBody/component/section/entry'
--    'consumable/manufacturedProduct/manufacturedMaterial'
-- are left out to fit informative parts.
--
-- ATC¦A10BA02¦Metforminum,.../substanceAdministration/.../code
-- ATC kood¦J07AN01¦BCG VACCINE SSI,
↪  .../procedure/entryRelationship/substanceAdministration/.../code
```

Listing 3. Example query on extracted OID information.

4.4.3 CDA XML Structure Probing via Section Paths

Our parser is document type agnostic; it does not care whether we process inpatient or outpatient discharge summaries, referrals, or responses to referrals. However, it parses different sections into different tables, e.g., section *Allergy* with marker `ALL` is parsed into tables `allergy` and `allergy_entry` (Figure 3).

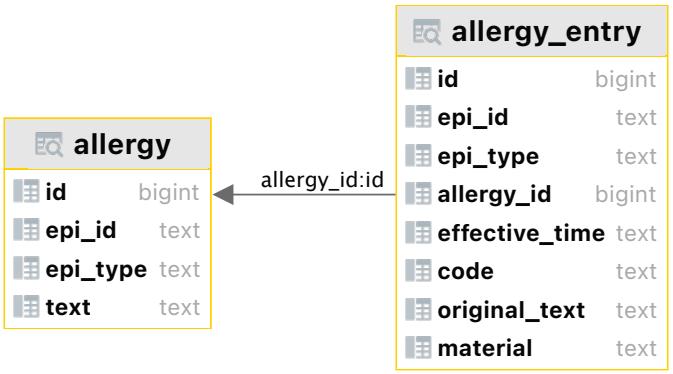


Figure 3. Relational database structure for allergy section

During OID extraction (see section 4.4.2, CDA XML Structure Probing via Object

Identifiers), we capture XPaths, but we still do not know in which sections particular items are located. In order to get a good overview of this, a further extension of our XSLT stylesheet was needed.

We added logic, that captures a section marker (i.e., `section/code/@code` value) for each record referencing values from a section. Taking into account that sections in CDA documents can also contain subsections, we capture the whole path to a particular section in a style `-SECTION-SUBSECTION-. . . -CURRENT_SECTION-`.[29]

Our newly extracted information allows us to get quick insights, such as overviews of the relative occurrence of specific sections (see table Table 1 in Appendix II. Top-level Section Types in CDA Documents), find out where ATC codes are used (see notes about ATC in Estonia) or that LOINC codes can also be found in `PAT_PROC` *(Pathology examination procedure)* sections.

4.4.4 Additional Explorability Options

Having built a coarse-grained net around our data (see CDA XML Structure Probing via Section Paths), we now know what kind of information is where, generally. Further extraction would need a closer inspection of XML section content, i.e., additional options for increasing explorability are still very much desired.

Extracting Structured Body Sections. We have prototyped lifting `<section>` part of XML into PostgreSQL and inspecting information further there, with special XML functions[30] provided by the database engine, operating on values of type `xml`.[31]

Our experiments show that leveraging PostgreSQL's `xml` type works for querying, as long as one is satisfied by the power provided by XPath 1.0.[32] To give an idea how this kind of exploration looks like, here are some example queries:

- Find all `<content>` elements under `list/item/` containing string `epigastrium`:

```
SELECT (xpath('//list/item/content[text()[contains(.,''epigastrium'')]]',
↪  section_xml))::text
-- {"<content>
```

[29] The current standard for section coding lists more than 120 section types, see http://pub.e-tervis.ee/classifications/Sektsiooni%20kodeering; according to that, it is safe to use - as a separator

[30] See https://www.postgresql.org/docs/current/functions-xml.html

[31] See https://www.postgresql.org/docs/current/datatype-xml.html

[32] See https://www.postgresql.org/docs/current/xml-limits-conformance.html

```
-- 12.07.2018 11:15<br/>Objektiivselt: valu paremal
↪  &lt;b&gt;epigastriumis&lt;/b&gt;.
-- </content>"}
```

- Find sections where we have line break elements like
,
 somewhere:

```
SELECT section_xml
FROM anamnesis_data
WHERE section_xml::text ILIKE '%<BR%';
```

Further Extraction. Naturally, question arises—if we can query and filter out subsets of XML then are those methods also suitable for further data transformation jobs?

Indeed, there exists `xmltable` expression to create records straight from the `xml` type values. Still, dealing with mixed content is tricky (again), as is handling of namespace prefixes. Notably, to preserve as much information as possible, it is advisable to keep the content type for extracted sub-parts as `xml`, and then cast it to `text` when the final cleaning phase arrives. However, the price paid here is the need to deal with XML markup text and entities like `<` and `>`. Essentially this means that in case of following sequence we do not lose mixed content item
 but XML will be kept, *as is*, as well:

1. A doctor marked an important part with bold:

   ```
   Objektiivselt: valu paremal epigastriumis.
   ```

2. An information system added time information and a line break:

   ```
   12.07.2018 11:15
   Objektiivselt: valu paremal epigastriumis.
   ```

3. The CDA document carried information as XML:

   ```
   <content>
   12.07.2018 11:15<br/>Objektiivselt: valu paremal
   ↪  &lt;b&gt;epigastriumis&lt;/b&gt;.
   </content>
   ```

The last part could be extracted also as

```
12.07.2018 11:15Objektiivselt: valu paremal <b>epigastriumis</b>.
```

but in this case we would lose information (a line break indicator).

4.5 Lessons Learned

- The reasons for creating a new parser are informative—you can use a lightweight tool for quick prototyping, but at some point, you have to make a tough decision and decide if it is good enough to meet all your needs. To avoid major rewrites, it is advisable to take time off to consider options sooner rather than later.
- An alternative to starting with lightweight tools is to start with a tool that is known to be robust and is already an industry standard.
- At present, getting an overview of the content of CDA documents or simply querying them is still more problematic than it should be. Here, one solution that could be tried would be to use dedicated XML databases that allow XML-specific queries.

5 Building Reusable Transformations

> If our algorithms were so magical that we could throw in raw text logs and real insights would fall out, guess what? We'd never clean.
>
> *Randy Au, in a blog post titled 'Data Cleaning IS Analysis, Not Grunt Work'*

This chapter discusses challenges associated with cleaning CDA data. It looks at the specific details essential for building reusable transformations to lift out the signal from the machine-processable, more structured parts (5.1, Structural Data) and narratives, i.e., free text parts (5.2, Fact Extraction).

5.1 Structural Data

Applying the CDA XML Parser to the CDA documents creates around 40 tables, each with various amount of fields and very different content, stored as PostgreSQL `text`.

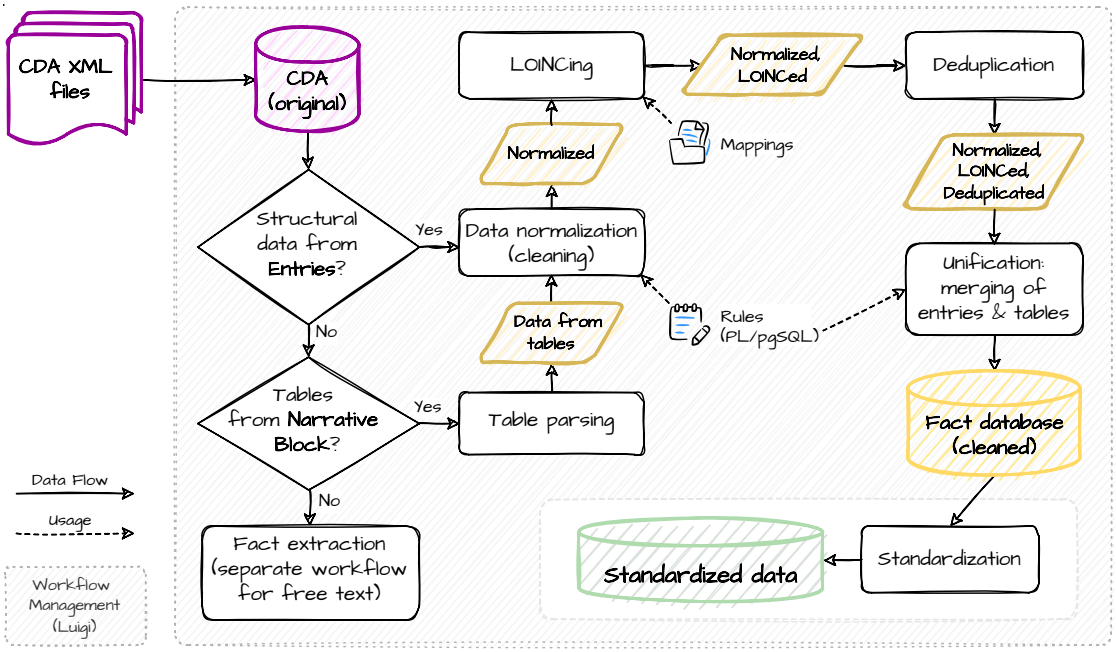


Figure 4. Structural data cleaning workflow

Cleaning the fields takes different paths depending on the content (Figure 4).

Data Normalization. In the case of machine-processable part from `<entry>` elements, we can start with data normalization, e.g., turn reference values of analyses into the same form, examples being:

- `1,8 .. 7,8` normalized to `[1.8,7.8]`,
- `5.0-8.0` normalized to `[5.0,8.0]`.

This normalization part relies on cleaning rules described as PL/pgSQL scripts employing various regular expressions. Given rules split values and units, detect and normalize values according to the actual underlying type (integer, float, range, ratio, timestamp) and dissect more complicated constructs like ranges with units (e.g., `9.8 10g/l-19.8 10g/l`) or texts with values (e.g., `Neg(4..5)`). We use pgTAP[33]-powered unit tests to keep all this complexity under control.

Table Parsing. In case of analyses, there are also XHTML-like tables under Narrative Block, inside `component/section/text` elements. Those tables need separate parsing to extract values from row cells and find correct context information (e.g., analysis date) from the table header. Extracting information from tables presented under Narrative Block is a significant undertaking because *how exactly* to present information there is left to be the decision of the implementer. Different information systems produce very different tables—in total, we have to deal with seven major types, each of them having several sub-types.

LOINCing. After the data normalization step, we can take up the challenge to assign LOINC codes, e.g., to differentiate analyses performed. This step, called *LOINCing*, relies on thousands of manually curated mappings that help find the correct LOINC code. For example, having extracted information that the analysis name is `Valk` (*Protein*), the parameter name is `S-Prot`, and the unit is `g/L`, we can map it to the LOINC code `2885-2`[34] with Long Common Name `Protein [Mass/volume] in Serum or Plasma` and Estonian Fully-Specified Name `Valk:MCnc:Pt:S/P:Qn:`, described in Estonian

[33] See `https://pgtap.org/`
[34] See `https://loinc.org/2885-2/`

Linguistic Variant with related names `Juhuslik Kvantitatiivne Plasma Seerum Seerum või plasma`.

Deduplication and Unification. To produce a clean fact database, we need to deduplicate our normalized, LOINC-enriched records and perform a merge of the information coming from different parts of the CDA document. The merging step handles different matching levels (perfect matches, unique matches, various combinations of compared fields, ties) and is implemented as a set of Luigi tasks relying on SQL scripts.

Created clean fact database is now the new basis for further standardization, e.g., mapping this information onto the OMOP CDM database (see chapter 6, Data Conversion onto OMOP Common Data Model).

5.2 Fact Extraction

Narrative parts of the CDA document can, besides tables, also contain other constructs to present information:

- Printouts of analyzes (Listing 4) or objective findings, e.g.:

```
Pikkus:  176    cm
Kaal:    95kg
VÜ  107  cm
KMI   30,7      lisaterviseriskiga ülekaal alates 93 kg
```

- Free text narratives describing medical history (anamnesis), can be assembled from different sub-parts prefixed with texts like `Kaebused:` (*Complaints*), `<DD/MM/YYYY> analüüsid:` (*Analyses*, with exact date), `Allergia:` (*Allergies*), `Operatsioonid:` (*Operations*). See Listing 5 for an example of an extract.

Dealing with free text, i.e., finding specific subsections and extracting dated events and facts, would be harder to tackle without dedicated Natural Language Processing (NLP) tools. We employ EstNLTK, a unified programming interface for everyday NLP tasks in the Estonian language, including state-of-the-art components for fact extraction [LOST20, Est22].

Our aim has been to create and apply reusable transformations to extract information about medications taken, various scores, results of additional analyses, measurements,

```
...
ANALÜÜSIDE TELLIMUS nr: <ANONYM id="8" type="landline" />

Märkus: Määratud 2 korda!

MATERJAL:
YY02853672 15.11.2013 10:54 (võetud: 15.11.2013 00:00)

VASTUSED:
Hemogramm
WBC 8.50 (3,5 .. 8,8 E9/L )
RBC 4.54 (3,9 .. 5,2 E12/L )
HGB 133 (117 .. 153 g/L )
HCT 37 (35 .. 46 % )
...
```

Listing 4. Example of a printout (partial output, dates and values modified)

```
...
Kaebused: Aeg-ajalt esinev valu rindkere keskosas (varasemalt diagnoositud
söögitoru põletikku).
Hommikune liigesjäikus 1 h.
Käesolev probleem: Hospitaliseeritud VI Rituksimabi infusiooniks.
Patsient tunneb ennast üsna hästi, hommikuti liigesjäikust 1 h, sel ajal tihti ka liigeste valulikkust,
↪  ägenemisi ei  ole olnud.
24/03/2014 analüüsid:
ASAT 26 U/L, ALAT 23 U/L, SR 7 mm/h, CRP 14,2 mg/L, Glükoos 6,2 mmol/L, kreatiniin 79 mikromol/L
WBC 3,7 x 10E3/mikroL, RBC 5,3 x 10E6/mikroL, HGB 13,6 g/dL, PLT 217 x 10E3/mikroL.
Hiljuti olnud haigestumine (kõhuviirus) eelmise nädala keskel (19.-21.03.2014), millest nüüdseks
↪  paranenud.
...
```

Listing 5. Example of an anamnesis (partial output, dates and values modified)

and indicators that are not yet strictly standardized for CDA document types. Out of those transformations, we have created a free text data cleaning workflow (Figure 5).

Texts. The workflow, orchestrated by a Luigi workflow management agent [LUI18], starts out by creating an EstNLTK collection of texts (searchable JSON objects) from select fields of select tables created via CDA XML Parser (see chapter 4, CDA Document Parsing). This adheres to the guidance for iterative development from EstNLTK authors [LOST20], stating: "The best way to organize the development of a fact extraction pipeline is to store all the texts as an ESTNLTK collection and then iteratively develop the set of necessary taggers."

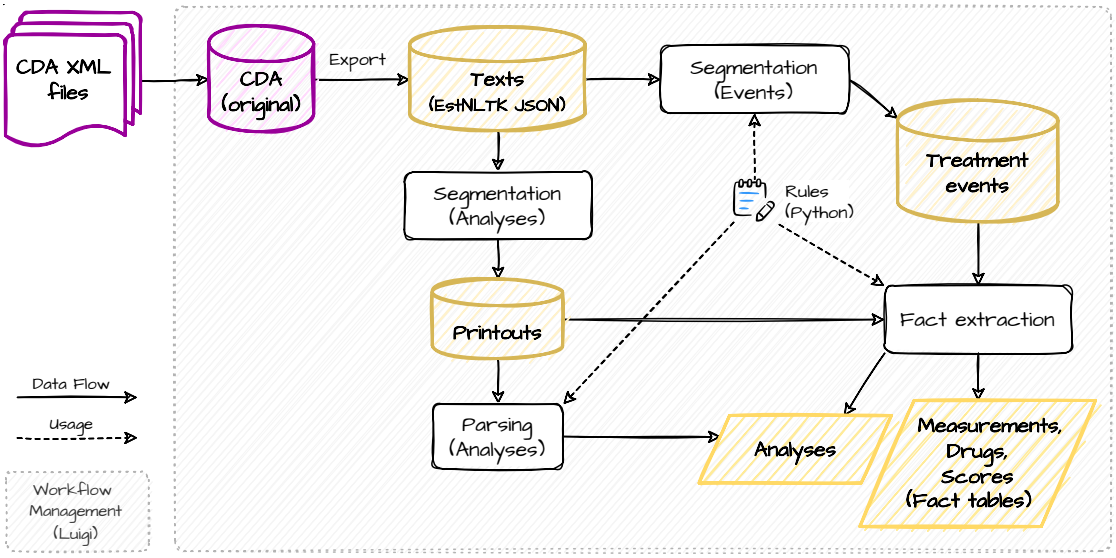


Figure 5. Free text data cleaning workflow

Segmentation. Next, to prepare for fact extraction, texts are segmented and handled a bit differently depending whether we are going to deal with analyses or other dated events. Analyses are mostly in printout form (see Listing 4 above), on those are applied dedicated `RegexTagger`-based EstNLTK taggers. Current taggers target seven major types of printouts [Ott21]. Other taggers segment out treatment events based on described start- and end-markers (described via special taggers on their own).

Fact Extraction. Final fact extraction step in this workflow pulls out specific values and creates fact tables for measurements, drugs, scores and analyses. After the final step facts are ready for additional cleaning (currently not yet performed).

5.3 Lessons Learned

- Creating reusable transformations to clean data is hard work. Managing them without proper data tests is even more difficult. We recommend looking for ways to create data tests early and manage them consistently.
- It is recommended to establish a proper continuous integration (CI) workflow for all essential work/tasks and eliminate the rest to reduce noise and confusion. Learn efficient repository branch management when extensive prototyping is required.

6 Data Conversion onto OMOP Common Data Model

We have now covered how to locate areas of interest in CDA documents (Section 4.4.2, Section 4.4.3), extract machine-processable and free text parts with the CDA XML Parser (Section 4.3.2), and build out and apply reusable transformations (Section 5) to arrive at clean fact database (Figure 6).

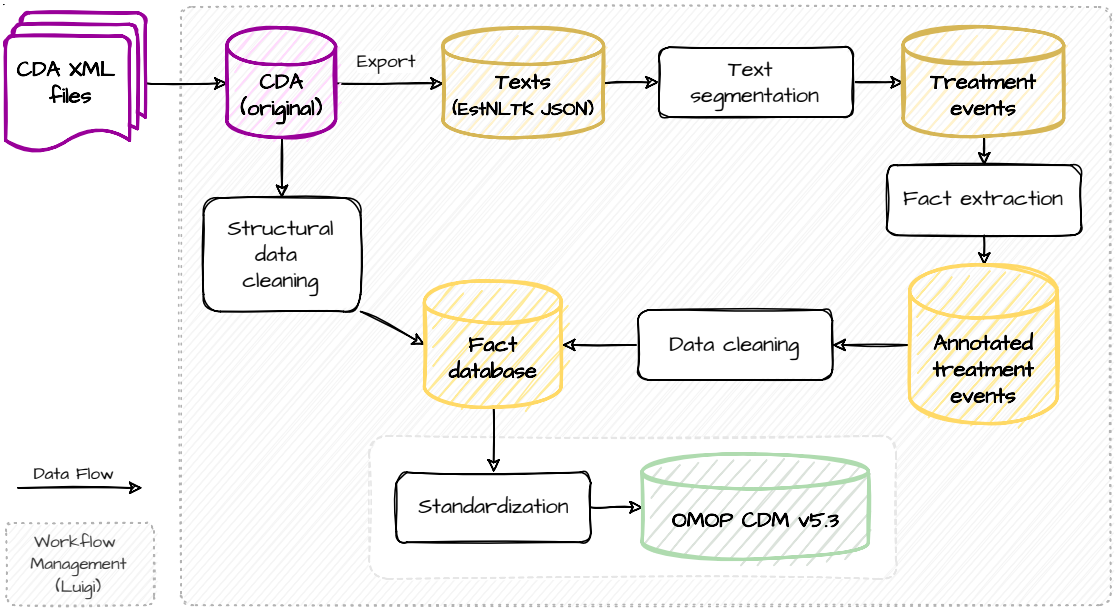


Figure 6. CDA document processing overview

Our next target is to standardize it further and move this data onto an OMOP CDM database designed to support a wide range of observational research activities [OHDSI20].

The following sub-section motivates and briefly describes the Standardized Vocabularies, a foundational resource and an integral part of the OMOP CDM, knowledge and understanding of which is required to do the standardization. After that, we will look into the approach we have used to handle mappings and perform the data conversion onto the CDM.

6.1 Mapping to Standard Vocabularies

6.1.1 Mapping Vocabularies – Why Caution is Warranted

Considering Estonian LOINC implementation, besides using official LOINC codes, analyses can be linked to two different Estonian-specific codes:

- `A-NNNN` code, a temporary code, which will be replaced by an official LOINC code over time (there is a promise or perceived intention for that);
- `L-NNNN` code, signifies local analyses which are not replaced by LOINC codes, as local analyses can be very laboratory specific [Pal21].

Analyses with local codes are mostly panels, but can also be simple analyses. An example of Estonian-only temporary code would be `A-4823`[35] which

- in local information system is titled as `LOINC #`,
- has local official designation as `eGFR (Crea, CKD-EPI, POCT)`,
- with full name being `Hinnanguline glomerulaarfiltratsiooni kiirus (kreatiniin, CKD-EPI, POCT)`.

Such a temporary, local code shall not be confused with a similar-looking official LOINC code, e.g. with a code `4823-1`[36] having textual labels like

- a Long Common Name `HLA-B35 [Presence]`,
- a Display Name `HLA-B35 Ql (Bld/Tiss)`, and
- a Fully-Specified Name `HLA-B35:PrThr:Pt:Bld/Tiss:Ord:`.

In the example above, one of those laboratory tests detects a substance capable of stimulating an immune response. In contrast, the other can measure how well one's kidneys work – totally different things.

The same prudence in telling differences apart applies when considering deceivingly similar LOINC Component names and other LOINC Parts [Vre16]. Thus, caution is

[35] See `https://elhr.digilugu.ee/data/labori_uuringudDetail.html?id=39727&loinc=A-4823` for details (in Estonian)

[36] See `https://loinc.org/4823-1/`

warranted when moving from local nomenclature to the official standards. Fortunately, in some cases, mapping activities are supported by specific guidance, like LOINC Technical Briefs [LLC22].

6.1.2 Representing Events and Facts via Standardized Vocabularies

In the OMOP Standardized Vocabularies, all events and administrative facts are represented as concepts, concept relationships, and concept ancestor hierarchy (Figure 7).

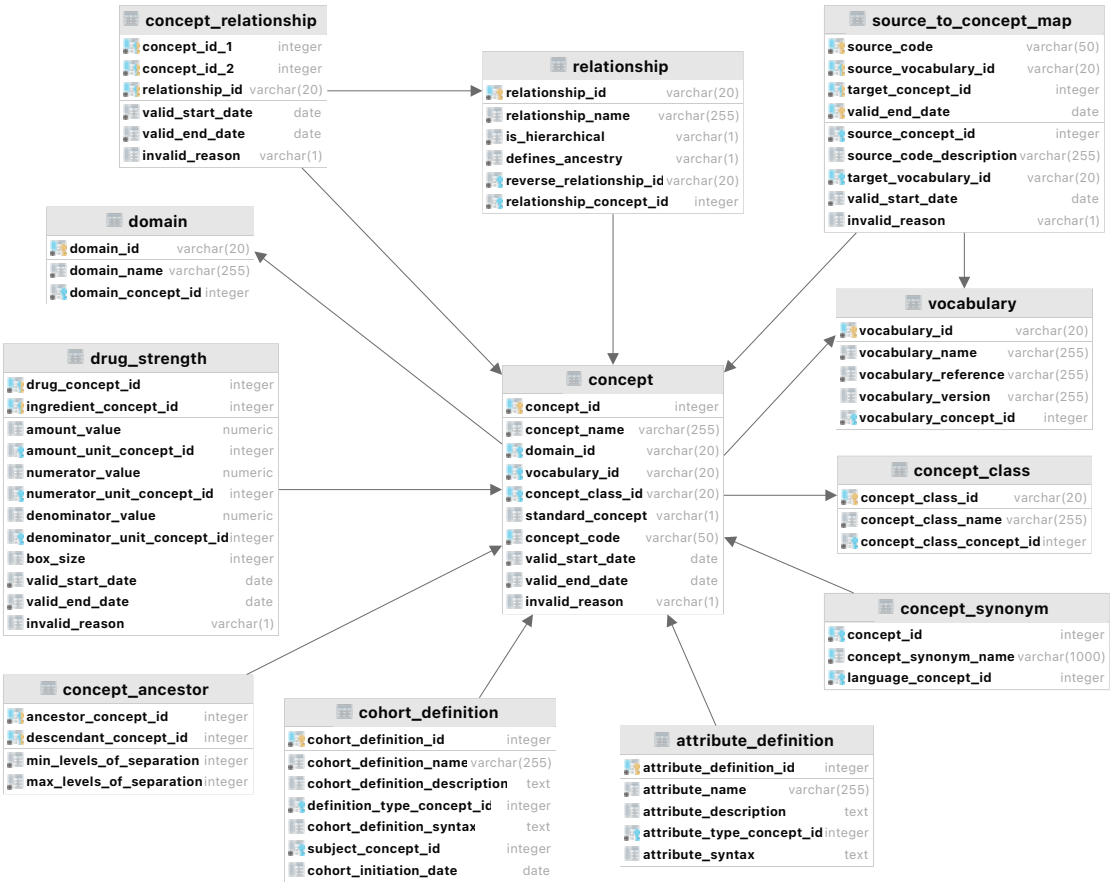


Figure 7. Database tables to manage Standardized Vocabularies (OMOP CDM v5.3)

Most of these concepts are adopted from existing coding schemes or vocabularies, while some of them are curated *de novo* by the OHDIS Vocabulary Team. The most extensive such OHDSI-curated domain vocabulary is RxNorm Extension, covering drugs not available on the US market [DOR17, OHDSI19]. As it turns out, it is also vital for

mapping medicines found in Estonian CDA documents.

Concepts of equivalent meaning in different vocabularies are mapped to one of them, designated the Standard Concept, indicated through an `S` in the `concept` table, the `standard_concept` field (Figure 7). Other concepts are source concepts. There is an additional class of concepts called *classification concepts*, which are non-standard, but in contrast to source concepts, they participate in the hierarchy [OHDSI20]. Let us look at some examples.

- An ICD-10 code `R19.3`[37] (*Abdominal rigidity*) is a non-standard source concept mapped to SNOMED CT code `72300008` (also named *Abdominal rigidity*), a standard concept with `concept_id` of `200215`.[38]
- An ATC code `A01AB09` (*miconazole; local oral*) is a classification concept (non-standard), mapped to the standard concept with `concept_id` of `907879` representing a RxNorm code `6932` (*miconazole*). Another ATC code (mentioned as an example in Section 2.2.1), `D01AC52` (*miconazole, combinations; topical*), also maps to the same standard concept.[39] However, as it is underspecified, it has many other relationships relating to various clinical drug forms, clinical packs, ingredients, and marketed products.

Each concept is assigned a domain identifier like *Condition*, *Drug*, *Procedure*, *Visit*, *Specimen*, and others. In total, OMOP CDM specifies more than 40 such domains. The domain controls where the fact (clinical event or event attribute) represented by the concept is stored in the CDM. Thus, information about *Abdominal rigidity* is stored at `condition_occurrence` (*Condition*), and information about *miconazole* use is stored at the `drug_exposure` table (*Drug*).

6.2 Data Conversion

6.2.1 General Flow of an ETL

Conversion of cleaned facts retrieved from CDA documents involves mostly

- finding out the right concepts for source codes,

[37] See https://icd.who.int/browse10/2019/en#/R19.3
[38] See https://athena.ohdsi.org/search-terms/terms/200215
[39] See https://athena.ohdsi.org/search-terms/terms/21601937

- directing them to the right CDM table according to the target domain, and
- adding additional information.

For example, in the case of drugs, we can record

- the start date of the exposure to the drug (`drug_exposure_start_date`),
- the last day on which the patient was still exposed to the drug, can be inferred from supply information (`drug_exposure_end_date`),
- the number of days of supply of the medication as prescribed (`days_supply`),
- the quantity of drug recorded in the original prescription or dispensing record (`quantity`).

Additionally, we need to find out the corresponding `concept_id` for the route of administration of the drug the patient was exposed to and capture source values (e.g., `A01AB09` → `drug_source_value`, `suukaudne` → `route_source_value`). Note that the current ETL adds only dispensed drug data and does not add prescribed drug data.[40]

OHDSI community suggests mapping the `person` table first, as this makes sense because the CDM is a person-centric model [OHDSI20]. Before mapping clinical event tables like `condition_occurrence`, `drug_exposure`, or `procedure_occurrence`, one should first concentrate on `observation_period` and `visit_occurrence`. After that, it is up to the personal preference which CDM tables to map and in which order.

6.2.2 Implementing Data Conversion

OHDSI community does not make a formal recommendation on implementing an ETL. The community has tried to build the ultimate user-friendly ETL tool in the past. After several independent attempts, it has been decided to give up on this [OHDSI20]. Usually, such generalized tools work well for 80% of the ETL, leaving the rest for gnarly workarounds and low-level code.

There are several examples of approaching an ETL for converting source data to CDM (commonly named *builders*). Such tools have been built in different languages like R, or

[40] Information about dispensed drugs is generally not available in CDA documents but can be acquired from prescription data governed by the Estonian Health Insurance Fund (EHIF)

C# + TSQL, are running on different runtimes like .NET[41] or compute services like AWS Lambda,[42] and employ different database engines like MS SQL Server, PostgreSQL, or MySQL [Con22b, Dev22, Con22a].

Our Approach on ETL. Our approach for converting data into CDM format uses SQL, specifically `psql` scripts coordinated via GNU Bash flavor shell scripts. Post-ETL steps, like generating descriptive statistics on an OMOP CDM database via Achilles [DRS+22] or running a series of data quality checks against an OMOP CDM instance via Data Quality Dashboard [DQD22], require the usage of R.

Infrastructure. Our instances of OMOP CDM v5.3 work on top of PostgreSQL 13 on an OpenStack-based VM cloud running Debian 10. The secure cloud environment is provided by the University of Tartu High Performance Computing Center (UTHPC) [HPC22], relying on the Estonian Scientific Computing Infrastructure (ETAIS) [ETA22].

Contribution. Coordinating the creation of initial databases with suitable configurations, role hierarchies, privileges, and OMOP CDM specific schemas supporting over time the work of about two dozen students and researchers has been a significant contribution of the author.

6.3 Lessons Learned

- The same principles as described previously apply to the maintenance of newly created cleaning transformations as well—use CIs and data tests.
- Push additional cleansing transformations upstream as soon as possible—this way, cleaning is done where needed, and less maintenance is required.

[41] See https://dotnet.microsoft.com/
[42] See https://aws.amazon.com/lambda/

7 Analytics Platform Deployment Automation

We have now seen what it takes to extract granular and less-granular parts from CDA documents (Section 4), how reusable transformations help us to extract cleaned facts (Section 5), and how to map it to the standardized OMOP CDM format (Section 6). Tooling requirements for those steps can be stated quite simply. We need the following:

- Debian 10 (although the latest CentOS or Ubuntu versions are equally suitable);
- Linux utils like GNU Bash, GNU Find Utilities (`find`, `xargs`), GNU parallel [Tan18], `xsltproc`, `git` and others;
- PostgreSQL 13 cluster with a client (`psql`);
- Miniconda with `conda` package manager;
- R, version 4+.

Tooling like this would suffice to run a study package (that needs only R) against our OMOP CDM database instance. However, implementing a study more interactively or relying on open-source R packages developed by the OHDSI community needs more environment tuning.

The following subsection gives a brief overview of open source and free analytics tools the OHDSI community offers. We also describe involved technologies, as this is important from a deployment point of view. A section after that describes our approach for getting the whole tool stack up and running.

7.1 OMOP Common Data Model Analytics Tools

7.1.1 ATLAS and WebAPI

ATLAS is a free, publicly available, web-based tool developed by the OHDSI community that facilitates the design and execution of analyses on standardized, patient-level observational data in the OMOP CDM format [OHDSI20]. It allows anyone in the OHDSI community to collaboratively design high-quality observational studies and produce reproducible code that can be shared as R packages and executed on OMOP CDM databases worldwide.

One can get a first-hand experience with ATLAS via its publicly available demo application at https://atlas-demo.ohdsi.org, employing Medicare Synthetic Public Use Files (SynPUF) simulated datasets of different sizes [DVD+15, CfMS21]. While the datasets are synthetic, it is still possible to define an effect estimation or prediction study and even generate the R code for executing the study.

The interactive experience of the ATLAS is built via HTML, CSS and a JavaScript library called Knockout.[43] However, heavy lifting behind the scenes is done by OHDSI WebAPI, a Java application running typically on the Apache Tomcat application server. OHDSI WebAPI contains all OHDSI services that can be called from OHDSI applications. The primary interaction flow is that ATLAS makes HTTP requests to the endpoints like https://api.ohdsi.org/WebAPI/info and gets back responses containing strings in a JSON format (e.g., containing sub-parts like "version":"2.11.0").

7.1.2 Libraries for Large Scale Analytics

OHDSI offers a set of open-source R packages that can be used to perform a complete observational study [OHDSI20].

This collection of packages, named HADES (Health Analytics Data-to-Evidence Suite), can provide standardized analytics for population characterization, population-level effect estimation, and patient-level prediction [HAD22]. While the collection's name is new (previously *OHDSI Methods Library*), it has been used in many published clinical and methodological studies.[44]

Some of the components in the OHDSI R packages in HADES require Java, e.g., to connect to the database. Usage of RStudio [All12] is suggested to obtain a pleasant R experience. Also, some of the analytics can be computationally intensive. Thus, having enough processing cores and plenty of memory is helpful; at least four cores and 16 GB of memory are recommended.

[43] See https://knockoutjs.com/

[44] See https://ohdsi.github.io/Hades/publications.html

7.2 Deployment Automation

> Build/CI systems are perhaps the most under-appreciated part of the practice of computer science. There is absolutely no shame in spending a week just working on the build. Unfortunately, there's no glory in it either.
>
> *Alexander Ioffe*

In general, we want to end up with a situation where our whole transformation pipeline and all the OHDSI analytics tools live nicely side-by-side to support tight iterative work on all of them without high context switching costs. By the current practice, we deploy the whole analytics environment as a single Linux virtual machine on ETAIS infrastructure (Figure 8) [ETA22, HPC22].

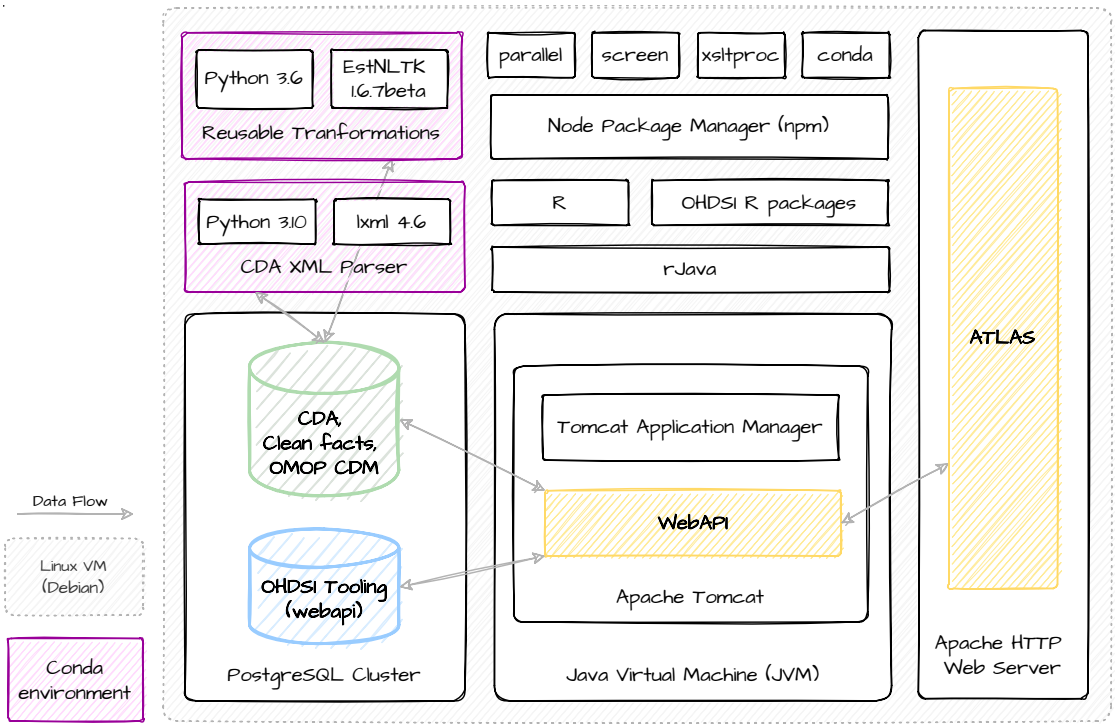


Figure 8. Components of analytics environment

7.2.1 Existing Solutions for Integrated Deployment Strategies

As noticed by the OHDSI community, deploying the entire OHDSI tool stack with ATLAS and HADES can be a daunting task [OHDSI20]. To lessen the burden, the community has developed different pre-packaged virtualization solutions.

Two solutions rely on Amazon AWS—OHDSI-in-a-Box[45] and OHDSIonAWS.[46] Neither of those is suitable for us, as we do not want to ship Estonian health data to the servers outside Estonia. Then there is an OHDSI Broadsea[47] that packages OHDSI tools into a single binary, a Docker container image. Unfortunately, it is not up-to-date in all aspects, and it is hard to change it to use specific versions of different tools or accompany the latest fixes in great need but not officially released yet.

7.2.2 Automating with Shell Scripts and Ansible

Over time, we have developed PSQL and shell scripts that set up our main database with role hierarchies, privileges and schemas suitable for initial CDA document data ingestion and further transformations.

Our first attempts at setting up full OHDSI stack extended this approach to the point where it was possible to set up the stack via invoking couple of separate steps:

- update and upgrade Debian, set locale, install `git`;
- set machine timezone, install various utilities, create and mount more spacious volume for data, install PostgreSQL so that it uses new data volume for storage, take care that PostgreSQL uses SCRAM authentication;[48]
- install AdoptOpenJDK, Maven, Tomcat, NodeJS, Apache HTTP Server;
- set up a separate database for WebAPI;
- build a WebAPI and deploy it to the Tomcat;
- build an ATLAS and deploy it to the Apache HTTP Server.

[45] See https://github.com/OHDSI/OHDSI-in-a-Box
[46] See https://github.com/OHDSI/OHDSIonAWS
[47] See https://github.com/OHDSI/Broadsea
[48] See https://www.postgresql.org/docs/current/auth-password.html

Provided that dozens of configuration files were correct, and items in templates were replaced with correct values, everything worked out in the end—the OHDSI stack got up and running, and served well studies on the RITA-MAITT dataset.

Half-connected Setup Routines. However, there were too many steps to execute and too many opportunities to fiddle with configurations to call this process repeatable. While all such worries are possible to automate away given enough time, tries, and repeated tests, it does not take away the pain that one has to log in to the machine operated and perform activities there.

Introducing Ansible. A more declarative approach with Ansible, an agentless general-purpose IT automation platform, has proven to be a solid replacement [Gee17]. With many existing modules to choose from and descriptive language based on YAML[49] and Jinja templates,[50] it empowers setup management in a way that seems more maintainable and extendable. The main difference is that a shell script is a set of instructions, while an Ansible playbook (a metaphor to describe Ansible's configuration files) is a description of desired state.

Ansible provides reliable management of remote execution and encourages idempotency,[51] i.e., Ansible deploys the same configuration to a server multiple times without making any changes after the first deployment. Also, note that Ansible allows running any shell command, and this feature makes it comfortable to move over from shell scripts. However, due to this, Ansible *cannot guarantee* idempotency.

Current State of Infrastructure Automation. We have managed to transform all the initial setup scripts in some form or another into Ansible playbooks. Some of them are used from those playbooks directly as SQL scripts or as templated SQL scripts. As multiple persons have tried to introduce automation for different operating systems (Debian, Ubuntu, CentOS), the final consolidation into a single integrated Ansible collection handling matters uniformly and without duplication is still a work in progress.

[49] See https://yaml.org/

[50] See https://github.com/pallets/jinja

[51] Idempotence is the ability to run an operation which produces the same result whether run once or multiple times [Gee17] (https://en.wikipedia.org/wiki/Idempotence#Computer_science_meaning)

Contribution. Creation of initial templates, shell, and SQL/PSQL scripts and pushing forward Ansible-based automation for Debian with secure and highly configurable PostgreSQL setup has been a significant contribution of the author.

7.3 Lessons Learned

- OHDSI tooling stack is nicely packaged for Amazon AWS; although other solutions exist, they might not be configurable and malleable enough for custom needs.
- Setting up an environment for the whole analytics platform for OMOP CDM-based research is not a small undertaking and shall not be approached lightly.
- Infrastructure setup automation pays off, eventually. Automation routines would be even more reliable and solid when exercised more often.
- Well-coordinated effort and focused goals regarding particular technology choices would result in faster convergence for the final solution.

8 Discussion

This chapter will present conclusions for this thesis. Furthermore, future work and perspectives are described.

8.1 Conclusion

This thesis aims to dissect steps that one needs to take to incorporate health information exchange data from Estonian CDA documents in observational health studies and propose enhanced solutions for some of them.

The main focus was on challenges around the initial information extraction step that needed better explorability over a given set of CDA documents and a more reliable and ergonomic parser for extracting machine-processable and narrative parts. Just as importantly, the thesis concentrated on the challenges of analytics platform deployment automation.

As a result of this thesis, better CDA XML structure probing tools and a 2nd-generation CDA XML Parser are available for the Health Informatics team at the University of Tartu. Additionally, an improved environment automation solution significantly speeds up the whole analytics platform setup. Previously, an effort to set up an environment for the whole extraction pipeline part could take days; now, it takes only minutes.

In addition, a detailed description of approaches for building reusable transformations to arrive at cleaned facts and treatment events is given in this thesis. Moreover, an overview of our approach for data conversion onto OMOP CDM is described. This can serve as a reference point for those who need to use or extend described transformations and a mapping pipeline to the CDM.

8.2 Future Work

> I've always been more interested in the future than in the past.
>
> *Grace Murray Hopper*

In future work, different direction can be explored further.

The overarching main emphasis should be on enabling research with Estonian CDA health data on small scale, with opportunities for large scale network studies. This warrants turnkey solutions to quickly set up an environment for every new dataset and related studies, with repositories supplying extraction, transformation, and OMOP-mapping workflows. Here, further automation as well as tuning of the workflows is needed.

Concerning CDA documents, we need to deal separately with several challenges. An issue of explorability could be solved using special-purpose XML databases or specially crafted solutions. A separate challenge is knowing where we are regarding our extraction and when we are *done*.

References

[AKL22] Andrew Atkinson, Krista Kärt, and Rutt Lindström. SNOMED CT Managed Service - Eesti laiendi versioonimärkmed - November 2021, 2022. URL: https://confluence.ihtsdotools.org/pages/viewpage.action?pageId=137234458 [cited 2022-04-06].

[All12] J.J. Allaire. RStudio: Integrated Development Environment for R. *Boston, MA*, 770(394):165–171, 2012.

[Ana22] Anaconda Inc. Miniconda, 2022. URL: https://docs.conda.io/en/latest/miniconda.html [cited 2022-05-08].

[ATC22] World Health Organization. Anatomical Therapeutic Chemical (ATC) Classification, 2022. URL: https://www.who.int/tools/atc-ddd-toolkit/atc-classification [cited 2022-04-21].

[BBS+16] Alison Bourke, Andrew Bate, Brian C Sauer, Jeffrey S Brown, and Gillian C Hall. Evidence generation from healthcare databases: recommendations for managing change. *Pharmacoepidemiology and Drug Safety*, 25(7):749–754, 2016.

[BFB+21] Stefan Behnel, Martijn Faassen, Ian Bicking, Holger Joukl, Simon Sapin, Marc-Antoine Parent, Olivier Grisel, Kasimier Buchcik, Florian Wagner, Emil Kroymann, et al. The lxml XML toolkit for Python, 2021. URL: https://github.com/lxml/lxml/tree/lxml-4.6 [cited 2022-05-08].

[BG21] Tim Benson and Grahame Grieve. *Principles of Health Interoperability*. Springer, 2021.

[Bha15] Subhra Bikash Bhattacharyya. *Introduction to SNOMED CT*. Springer, 2015.

[Bod04] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[BOD+21] Patricia Biedermann, Rose Ong, Alexander Davydov, Alexandra Orlova, Philip Solovyev, Hong Sun, Graham Wetherill, Monika Brand, and Eva-Maria Didden. Standardizing registry data to the OMOP Common Data Model: experience from three pulmonary hypertension databases. *BMC medical research methodology*, 21(1):1–16, 2021.

[Boo11] Keith W Boone. *The CDA™ Book*. Springer Science & Business Media, 2011.

[BSGC19] Chris Burr, Henry Schreiner, Enrico Guiraud, and Javier Cervantes. Sustainable software packaging for end users with conda. Technical report, CERN, 2019.

[C+97] Kona Proposal Committee et al. The Kona Proposal for Electronic Health Care Records, 1997. URL: https://www.hytime.org/ihc97/papers/harding/kona/kona.html [cited 2022-04-30].

[CfMS21] Centers for Medicare and Medicaid Services. Medicare Claims Synthetic Public Use Files (SynPUFs), 2021. URL: https://www.cms.gov/research-statistics-data-and-systems/downloadable-public-use-files/synpufs [cited 2022-05-13].

[Cim98] James J Cimino. Desiderata for controlled medical vocabularies in the twenty-first century. *Methods of information in medicine*, 37(04/05):394–403, 1998.

[Con22a] OHDSI/ETL-LambdaBuilder Contributors. CDM Builder leveraging AWS Lambda, 2022. URL: https://github.com/OHDSI/etl-lambdabuilder [cited 2022-05-15].

[Con22b] OHDSI/ETL-Synthea Contributors. Conversion from Synthea CSV to OMOP CDM, 2022. URL: https://github.com/OHDSI/etl-synthea [cited 2022-05-15].

[CST+20] Sylvia Cho, Margaret Sin, Demetra Tsapepas, Leigh-Anne Dale, Syed A Husain, Sumit Mohan, and Karthik Natarajan. Content coverage evaluation

of the OMOP vocabulary on the transplant domain focusing on concepts relevant for kidney transplant outcomes analysis. *Applied Clinical Informatics*, 11(04):650–658, 2020.

[DAB+99] Robert H Dolin, Liora Alschuler, Fred Behlen, Paul V Biron, Sandy Boyer, Dan Essin, Lloyd Harding, Tom Lincoln, John E Mattison, Wes Rishel, et al. HL7 Document Patient Record Architecture: An XML Document Architecture Based on a Shared Information Model. In *Proceedings of the AMIA Symposium*, page 52. American Medical Informatics Association, 1999.

[DAB+01] Robert H Dolin, Liora Alschuler, Calvin Beebe, Paul V Biron, Sandra Lee Boyer, Daniel Essin, Elliot Kimber, Tom Lincoln, and John E Mattison. The HL7 Clinical Document Architecture. *Journal of the American Medical Informatics Association*, 8(6):552–569, 2001.

[DAB+06] Robert H Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M Behlen, Paul V Biron, and Amnon Shabo. HL7 Clinical Document Architecture, Release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, 2006.

[Den21] James Densmore. *Data Pipelines Pocket Reference*. O'Reilly Media, Inc., 2021.

[Dev22] Janssen Research & Development. ETL-CDMBuilder: A .NET Core application to perform ETL to OMOP CDM for multiple databases, 2022. URL: https://github.com/OHDSI/etl-cdmbuilder [cited 2022-05-15].

[DOR17] Dimitry Dymshyts, Anna Ostropolets, and Christian Reich. International RxNorm Extension to Support the Expansion of the OHDSI Research Network Beyond the US, 2017. URL: https://www.ohdsi.org/web/wiki/lib/exe/fetch.php?media=resources:rxe_poster_2017.pdf [cited 2022-05-14].

[DQD22] OHDSI/DataQualityDashboard Contributors. Data Quality Dashboard: A tool to help improve data quality standards in observational data science, 2022. URL: https://github.com/OHDSI/DataQualityDashboard [cited 2022-05-15].

[DRS+22] Frank DeFalco, Patrick Ryan, Martijn Schuemie, Vojtech Huser, Chris Knoll, Ajit Londhe, Taha Abdul-Basser, and Anthony Molinaro. *Achilles: Generates descriptive statistics for an OMOP CDM instance*, 2022. R package version 1.7. URL: https://ohdsi.github.io/Achilles/.

[DVD+15] Mark D Danese, Erica A Voss, Jennifer Duryea, Michelle Gleeson, Ryan Duryea, Amy Matcho, Donald O'Hara, William E Stephens, Adler J Perotte, Lee Evans, et al. Feasibility of Converting the Medicare Synthetic Public Use Data Into a Standardized Data Model for Clinical Research Informatics. In *AMIA*, 2015.

[EEK+21] Elisabeth Eller, Gerli Eltermaa, Krista Kärt, Rutt Lindström, and Kady Sild. TIS laboriandmete teenuse loendite kehtestamine, muutmine ja kehtetuks tunnistamine, 2021. (Version 1, 2021-10-20). URL: https://pub.e-tervis.ee/manuals/TIS%20laboriandmete%20teenuse%20loendite%20kehtestamine%2C%20muutmine%20ja%20kehtetuks%20tunnistamine/1/TIS%20laboriandmete%20teenuse%20loendite%20kehtestamine%2C%20muutmine%20ja%20kehtetuks%20tunnistamine.pdf [cited 2022-04-08].

[Ees22] Eesti Laborimeditsiini Ühing. Loendid TEHIKu publitseerimiskeskuses, 2022. URL: https://www.elmy.ee/tooruhmad/loinc/loendid-tehiku-publitseerimiskeskuses/ [cited 2022-04-15].

[EHD22] EHDEN Consortium. The European Health Data & Evidence Network (EHDEN), 2022. URL: https://www.ehden.eu/ [cited 2022-05-15].

[Elk12] Peter L Elkin. *Terminology and terminological systems*. Springer Science & Business Media, 2012.

[Est22] EstNLTK contributors. EstNLTK – Open source tools for Estonian natural language processing, 2022. URL: https://github.com/estnltk/estnltk [cited 2022-05-04].

[ETA22] ETAIS Consortium. Estonian Scientific Computing Infrastructure (ETAIS), 2022. URL: https://etais.ee/ [cited 2022-05-15].

[FMD+96] Arden W Forrey, Clement J Mcdonald, Georges DeMoor, Stanley M Huff, Dennis Leavelle, Diane Leland, Tom Fiers, Linda Charles, Brian Griffin, Frank Stalling, et al. Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical chemistry*, 42(1):81–90, 1996.

[FSG+20] Patrick Fischer, Mark R Stöhr, Henning Gall, Achim Michel-Backofen, and Raphael W Majeed. Data integration into OMOP CDM for heterogeneous clinical data collections via HL7 FHIR bundles and XSLT. In *Digital Personalized Health and Medicine*, pages 138–142. IOS Press, 2020.

[Gee17] Jeff Geerling. *Ansible for DevOps: Server and configuration management for humans*. Leanpub, 2017.

[HAD22] OHDSI Community. Health Analytics Data-to-Evidence Suite (HADES), 2022. URL: https://ohdsi.github.io/Hades/ [cited 2022-05-15].

[HPC22] University of Tartu High Performance Computing Center. The High Performance Computing Center, 2022. URL: https://hpc.ut.ee/ [cited 2022-05-15].

[HRSG19] Andrea Haberson, Christoph Rinner, Alexander Schöberl, and Walter Gall. Feasibility of mapping austrian health claims data to the OMOP common data model. *Journal of Medical Systems*, 43(10):1–5, 2019.

[Int22] SNOMED International. SNOMED - Our Customers - Estonia, 2022. URL: https://www.snomed.org/our-customers/member/estonia [cited 2022-04-06].

[JKY+20] Hyerim Ji, Seok Kim, Soyoung Yi, Hee Hwang, Jeong-Whun Kim, and Sooyoung Yoo. Converting clinical document architecture documents to the common data model for incorporating health information exchange data in observational health studies: CDA to CDM. *Journal of Biomedical Informatics*, 107:103459, 2020.

[KAA+21] Isaac S Kohane, Bruce J Aronow, Paul Avillach, Brett K Beaulieu-Jones, Riccardo Bellazzi, Robert L Bradford, Gabriel A Brat, Mario Cannataro,

James J Cimino, Noelia García-Barrio, et al. What every reader should know about studies using electronic health record data but may be afraid to ask. *Journal of medical Internet research*, 23(3), 2021.

[LAAB+21] Antoine Lamer, Osama Abou-Arab, Alexandre Bourgeois, Adrien Parrot, Benjamin Popoff, Jean-Baptiste Beuscart, Benoît Tavernier, Mouhamed Djahoum Moussa, et al. Transforming Anesthesia Data Into the Observational Medical Outcomes Partnership Common Data Model: Development and Usability Study. *Journal of medical Internet research*, 23(10):e29259, 2021.

[Lan20] Lukas Lang. *Mapping eines deutschen, klinischen Datensatzes nach OMOP Common Data Model*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.

[LDD+19] Kristine E Lynch, Stephen A Deppen, Scott L DuVall, Benjamin Viernes, Aize Cao, Daniel Park, Elizabeth Hanchrow, Kushan Hewa, Peter Greaves, and Michael E Matheny. Incrementally transforming electronic medical records into the observational medical outcomes partnership common data model: a multidimensional quality assurance approach. *Applied clinical informatics*, 10(05):794–803, 2019.

[LDD+20] Antoine Lamer, Nicolas Depas, Matthieu Doutreligne, Adrien Parrot, David Verloop, Marguerite-Marie Defebvre, Grégoire Ficheur, Emmanuel Chazard, and Jean-Baptiste Beuscart. Transforming French electronic health records into the Observational Medical Outcome Partnership's common data model: a feasibility study. *Applied clinical informatics*, 11(01):013–022, 2020.

[Lin21a] Rutt Lindström. Review Tool for SNOMED CT Estonian Extension Release Files, 2021. (Diploma thesis, in Estonian). URL: https://digikogu.taltech.ee/en/Item/cbb30554-2d1d-44b0-bb09-5401b4b18f14 [cited 2022-04-06].

[Lin21b] Rutt Lindström. SNOMED CT in Estonia, 2021. (Slides). URL: https://thl.fi/documents/920442/8265631/SNOMED-In-Estonia-2021-12-08_Lindstrom.pdf [cited 2022-04-06].

[LLC22] Laboratory LOINC Committee. Choosing the Correct LOINC for Estimated Glomerular Filtration Rate, 2022. (Last updated: 2022-02-16). URL: https:

//loinc.org/kb/users-guide/loinc-technical-briefs/choosing-the-correct-loinc-for-estimated-glomerular-filtration-rate/ [cited 2022-04-20].

[LOST20] Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tammo. EstNLTK 1.6: Remastered Estonian NLP Pipeline. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7154–7162, Marseille, France, May 2020. European Language Resources Association. URL: https://www.aclweb.org/anthology/2020.lrec-1.884.

[LUI18] The Luigi Authors. Luigi Documentation, 2018. URL: https://luigi.readthedocs.io [cited 2022-05-13].

[McL18] Grant McLean. XML::Simple::FAQ – What isn't XML::Simple good for?, 2018. URL: https://metacpan.org/dist/XML-Simple/view/lib/XML/Simple/FAQ.pod#What-isn't-XML::Simple-good-for? [cited 2022-05-07].

[MHS+03] Clement J McDonald, Stanley M Huff, Jeffrey G Suico, Gilbert Hill, Dennis Leavelle, Raymond Aller, Arden Forrey, Kathy Mercer, Georges DeMoor, John Hook, et al. LOINC, a universal standard for identifying laboratory observations: a 5-year update. *Clinical chemistry*, 49(4):624–633, 2003.

[MLS+18] Christian Maier, L Lang, Holger Storf, Patric Vormstein, R Bieber, Johannes Bernarding, Tim Herrmann, Christian Haverkamp, P Horki, J Laufer, et al. Towards implementation of OMOP in a German university hospital consortium. *Applied clinical informatics*, 9(01):054–061, 2018.

[NDC22] U.S. Food & Drug Administration. National Drug Code Directory, 2022. URL: https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory [cited 2022-04-22].

[NZK+11] Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–448, 2011.

[OHDSI19] Observational Health Data Sciences and Informatics. RxNorm Extension - an OHDSI resource to represent international drugs, 2019. URL:

https://www.ohdsi.org/web/wiki/doku.php?id=documentation:international_drugs [cited 2022-05-14].

[OHDSI20] Observational Health Data Sciences and Informatics. The Book of OHDSI, 2020. URL: https://ohdsi.github.io/TheBookOfOhdsi/ [cited 2022-05-13].

[ORR+12] J Marc Overhage, Patrick B Ryan, Christian G Reich, Abraham G Hartzema, and Paul E Stang. Validation of a common data model for active safety surveillance research. *Journal of the American Medical Informatics Association*, 19(1):54–60, 2012.

[Ott21] Anne Ott. Medistiinitekstidest info eraldamine EstNLTK abil, 2021. (Project Report, supervised by Sven Laur).

[Pal21] Viljar Pallo. LOINC standardi juurutamise juhend, 2021. (Version 4, 2021-02-22). URL: http://pub.e-tervis.ee/manuals/LOINC%20standardi%20juurutamise%20juhend/4/LOINC%20standardi%20juurutamise%20juhend.pdf [cited 2022-04-20].

[PUB22] Tervise ja Heaolu Infosüsteemide Keskus. Standardite ja klassifikaatorite publitseerimiskeskus, 2022. URL: https://pub.e-tervis.ee/ [cited 2022-04-07].

[Reg22a] Regenstrief Institute, Inc. LOINC Term Basics, 2022. URL: https://loinc.org/get-started/loinc-term-basics/ [cited 2022-04-09].

[Reg22b] Regenstrief Institute, Inc. LOINC Users' Guide, 2022. URL: https://loinc.org/kb/users-guide/ [cited 2022-04-09].

[Reg22c] Regenstrief Institute, Inc. What LOINC is, 2022. URL: https://loinc.org/get-started/what-loinc-is/ [cited 2022-04-08].

[RxN22] US National Library of Medicine. RxNorm, 2022. URL: https://www.nlm.nih.gov/research/umls/rxnorm/ [cited 2022-04-16].

[SBM+04] Gunther Schadow, Paul Biron, Lloyd McKenzie, Grahame Grieve, and Doug Pratt. ANSI/HL7 V3 DT, R1-2004, HL7 Version 3 Standard: Data Types

— Abstract Specification, Release 1, 2004. URL: https://vico.org/CDAR22005_HL7SP/infrastructure/datatypes/datatypes.htm [cited 2022-05-08].

[SJD+21] Selva Muthu Kumaran Sathappan, Young Seok Jeon, Trung Kien Dang, Su Chi Lim, Yi-Ming Shao, E Shyong Tai, and Mengling Feng. Transformation of Electronic Health Records and Questionnaire Data to OMOP CDM: A Feasibility Study Using SG_T2DM Dataset. *Applied clinical informatics*, 12(04):757–767, 2021.

[SVR+22] Mihkel Solvak, Jaak Vilo, Sulev Reisberg, Sirli Tamm, Marek Oja, Kadri Ligi, Taavi Unt, Andres Võrk, Peeter Leets, Liina Kamm, Andre Ostrak, Hiroki Kaminaga, Triin Siil, Tanel Tammet, Risto Vaarandi, Sven Nõmm, Toomas Lepik, Veiko Lember, Steven Nõmmik, Colin van Noordt, Martin Ebers, Paloma Krõõt Tupay, Gaabriel Tavits, and Tanel Kerikmäe. Programmi RITA tegevuse 1 projekti „Masinõppe ja AI toega teenused“ lõpparuanne, 2022. URL: https://sisu.ut.ee/maitt/projektist?lang=et [cited 2022-05-09].

[Tan18] Ole Tange. *GNU Parallel 2018*. Ole Tange, April 2018. doi:10.5281/zenodo.1146014.

[TTC+20] Hui Xing Tan, Desmond Teo, Haroun Chahed, Cynthia Sung, Doreen Tan, Pei San Ang, and Sreemanee Raaj Dorajoo. Transforming electronic medical records to a common data model for real-world benefit-risk assessments in a tertiary care facility in Singapore. *Authorea Preprints*, 2020.

[Vab22a] Vabariigi Valitsus. Tervise infosüsteemi edastatavate dokumentide andmekoosseisud ning nende esitamise tingimused ja kord, 2022. (RT I, 04.02.2022, 7). URL: https://www.riigiteataja.ee/akt/104022022007?leiaKehtiv [cited 2022-04-15].

[Vab22b] Vabariigi Valitsus. Tervise infosüsteemi põhimäärus, 2022. (RT I, 29.01.2022, 15). URL: https://www.riigiteataja.ee/akt/106122016011?leiaKehtiv [cited 2022-04-15].

[Vre16] Daniel J. Vreeman. *LOINC Essentials – A step by step guide to getting your local codes mapped to LOINC*. Blue Sky Premise, LLC, 2016.

[WHO18a] WHO Collaborating Centre for Drug Statistics Methodology. ATC: Structure and principles, 2018. URL: https://www.whocc.no/atc/structure_and_principles/ [cited 2022-04-21].

[WHO18b] WHO Collaborating Centre for Drug Statistics Methodology. ATC/DDD methodology: Purpose of the ATC/DDD system, 2018. URL: https://www.whocc.no/atc_ddd_methodology/purpose_of_the_atc_ddd_system/ [cited 2022-04-21].

[ZMB+13] Xiaofeng Zhou, Sundaresan Murugesan, Harshvinder Bhullar, Qing Liu, Bing Cai, Chuck Wentworth, and Andrew Bate. An evaluation of the THIN database in the OMOP Common Data Model for active drug safety surveillance. *Drug safety*, 36(2):119–134, 2013.

Appendix

I. Glossary of Terms, Abbreviations, and Acronyms

American National Standards Institute (ANSI)

A private organization that oversees voluntary consensus standards in the United States.

Anatomical Therapeutic Chemical (ATC)

A pharmaco-epidemiologic classification system that groups the active medical substances according to the organ or system on which they act and according to their therapeutic, pharmacologic and chemical properties. Developed and maintained by the World Health Organization (WHO) Collaborating Centre for Drug Statistics Methodology in Norway.

biomedical informatics

The interdisciplinary field that studies and pursues the effective uses of biomedical data, information, and knowledge for scientific inquiry, problem solving, and decision making, driven by efforts to improve human health. *See also*: health informatics.

CDA document

A clinical document stored in the CDA format.

clinical document

A document typically produced by a clinician and documents clinical observations and services provided to a patient or subject of care.

Clinical Document Architecture (CDA)

An HL7 standard for naming and structuring clinical documents, such as reports.

The standard defines the structure and semantics of medical documents for the purpose of exchange. CDA documents are encoded in Extensible Markup Language

(XML). They derive their meaning from the HL7 Reference Information Model (RIM) and use the HL7 Version 3 Data Types, which are part of the HL7 RIM.

coding system

A collection of codes; also called a terminology or vocabulary. Each code in a coding system identifies a unique concept. Coding systems can be a simple list of terms that are not explicitly related to each other (e.g., LOINC), or they can be organized in a hierarchy (e.g., ICD-10), or through variety of different relationships (e.g., SNOMED CT).

In CDA documents the `codeSystem` attribute identifies a set (or namespace) of concepts; the `codeSystem` attribute must always be an OID, and must always be present.

See also: controlled terminology.

Common Data Model (CDM)

A convention for representing healthcare data that allows portability of analysis (the same analysis unmodified can be executed on multiple datasets). *See also*: OMOP CDM.

controlled terminology

A finite, enumerated set of terms intended to convey information unambiguously.

Defined Daily Dose (DDD)

The assumed average maintenance dose per day for a drug used for its main indication in adults.

The DDD is a unit of measurement and does not necessarily reflect the recommended or Prescribed Daily Dose. DDDs are only assigned to drugs with an ATC code and a DDD will normally not be assigned for a substance before a product is approved and marketed in at least one country.

Docker

An open platform for developing, shipping, and running applications; provides the ability to package and run an application in a loosely isolated environment called a

container.

See https://docs.docker.com/get-started/overview/ for more information.

e-Laboratory Management Application (eLHR)

A subsystem supporting the laboratory data service of the Estonian nation-wide Health Information System (EHIS), created in 2018 in TEHIK. The system allows laboratory administrators to manage their laboratory data independently, while the general administrator (*aka* LOINC administrator) is responsible for handling new analyses being added to the database, requesting new codes from the Regenstrief Institute, etc.

In Estonian: *e-labori haldamise rakendus (eLHR)*; available via https://elhr.digilugu.ee.

Electronic Health Record (EHR)

A repository of electronically maintained information about an individual's lifetime health status and health care, stored such that it can serve the multiple legitimate users of the record. *See also*: EMR.

Electronic Medical Record (EMR)

The electronic record documenting a patient's care in a provider organization such as a hospital or a physician's office. Often used interchangeably with Electronic Health Record (EHR), although EHRs refer more typically to an individual's lifetime health status and care rather than the set of particular organizationally-based experiences.

Estonian nation-wide Health Information System (EHIS)

Estonian central national EHR system with related information systems and services, operated by TEHIK. The system processes health-related data for the purposes of concluding and fulfilling a health service contract, ensuring the quality of health services and patients' rights, protecting public health and maintaining health status registers, health statistics and health care management.

Public interface to the system, the patient portal (https://www.digilugu.ee/)

has been available since the end of 2008.

In Estonian: *Tervise Infosüsteem (TIS)*.

Estonian Society for Laboratory Medicine (ELMÜ)

A union of laboratory physicians, specialists, clinical microbiologists, and other physical and juridical persons with an interest in the development of the laboratory medicine field. Formed in 1999; a full member of the International Federation of Clinical Chemistry and Laboratory Medicine (IFCC).

Among the activities of ELMÜ is the coordination of different workgroups, one of which is concentrating on the topic of the implementation of LOINC terminology.

Extensible HyperText Markup Language (XHTML)

A World Wide Web Consortium (W3C) recommendation, an XML syntax for HTML. As of now, this specification is superseded, new implementations should follow the latest version of the HTML specification at `https://html.spec.whatwg.org/`.

Extensible Markup Language (XML)

A subset of the Standard Generalized Markup Language (SGML) from the World Wide Web Consortium (W3C), designed especially for Web documents. It allows designers to create their own custom-tailored tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

Extensible Stylesheet Language Transformations (XSLT)

A declarative, pattern based language for transforming XML documents into other XML documents, or other formats such as HTML for web pages, plain text, CSV files.

The term *stylesheet* reflects the fact that one of the important roles of XSLT is to add styling information to an XML source document, by transforming it into a presentation-oriented format such as HTML, XHTML, or SVG. However, XSLT is used for a wide range of transformation tasks, not exclusively for formatting and

presentation applications.

XSLT is often used to display CDA documents, validate their content, or to transform other XML formats to CDA documents.

Extract, Transform, and Load (ETL)

ETL is the process by which source data is collected and manipulated so as to adhere to the structure and semantics of a receiving data construct, such as a data warehouse or batch processing system.

Health and Welfare Information Systems Centre (TEHIK)

Estonian national ICT competence center in the field of health, social security and labour in the administrative area of the Ministry of Social Affairs. TEHIK (in Estonian *Tervise ja Heaolu infosüsteemide Keskus*) takes care of 70 information systems and more than 100 services in health, labour and social fields all over the country.

Main responsibilities are development of information systems, databases and e-services, maintenance of services and infrastructure, providing information security and data analysis to support policy making, reporting productivity.

health informatics

Used by some as a synonym for biomedical informatics, this term is increasingly used solely to refer to applied research and practice in clinical and public health informatics.

health Information exchange (HIE)

The process of moving health information electronically among disparate health care organizations for clinical care and other purposes; or an organization that is dedicated to providing health information exchange.

Health Level 7 (HL7)

The standards developing organization based in the US, an ad hoc standards group formed to develop standards for exchange of health care data between independent computer applications; more specifically, the health care data messaging standard developed and adopted by the HL7 standards group.

healthcare organization (HCO)

Any health-related organization that is involved in direct patient care.

healthcare provider (HCP)

A healthcare professional or a legal entity providing healthcare services. *See also*: healthcare organization (HCO).

In Estonian: *Terviseteenuse osutaja (TTO)*.

healthcare service

A service provided for the promotion of physical and mental health, harm reduction, disease prevention, diagnosis and treatment.

information model

A representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. It can provide shareable, stable, and organized structure of information requirements for the domain context.

An information model describes the classes of information required and the properties of those classes including attributes, relationships, and states. Reference Information Model (RIM) by HL7 is an example of an information model.

interface terminology

Terms that allow users to interact easily with clinical concepts through common colloquial terms and synonyms. These terminologies generally embody a rich set of flexible, 'user-friendly' phrases.

International Classification of Diseases (ICD)

International standard for classifying diseases and other health problems recorded on health and vital records. The ICD is also used to code and classify mortality data from death certificates.

Developed by the World Health Organization (WHO).

JavaScript Object Notation (JSON)

A lightweight, text-based, language-independent syntax for defining data interchange formats. It was derived from the ECMAScript programming language, but

is programming language independent. JSON defines a small set of structuring rules for the portable representation of structured data.

The JSON syntax is not a specification of a complete data interchange. Meaningful data interchange requires agreement between a producer and consumer on the semantics attached to a particular use of the JSON syntax. What JSON does provide is the syntactic framework to which such semantics can be attached.

Logical Observation Identifiers Names and Codes (LOINC)

A controlled terminology created for providing coded terms for observational procedures. Originally focused on laboratory tests, it has expanded to include many other diagnostic procedures.

Developed by the Regenstrief Institute, a US nonprofit medical research organization.

Object Identifier (OID)

A globally unique ISO (International Organization for Standardization) identifier, a preferred scheme for unique identifiers in HL7.

OIDs are paths in a tree structure, with the left-most number representing the root and the right-most number representing a leaf. HL7 has chosen to represent an OID as a sequence of non-negative integers separated by periods (e.g., `2.16.840.1.113883.6.73` is an OID for ATC under `HL7 Registered External Coding Systems` (`2.16.840.1.113883.6`)).

HL7 provides a publicly available OID registry from which anyone can obtain an OID for their own use or look up OIDs used or assigned to others; see `https://www.hl7.org/oid/index.cfm`. Estonian-specific OIDs used in the scope of EHIS are managed by TEHIK and can be browsed via `https://pub.e-tervis.ee/oids.py`, background info is available at `https://www.tehik.ee/iso-oid`.

Observational Health Data Sciences and Informatics (OHDSI)

A multi-stakeholder, interdisciplinary, open-science collaborative to bring out the value of health data through large-scale analytics; pronounced "Odyssey".

With hundreds of researchers from 30 countries and health records for about 600 million unique patients from around the world, OHDSI seeks to improve health by empowering a community to collaboratively generate the evidence that promotes better health decisions and better care. OHDSI's data network is based on its OMOP Common Data Model, enabling federated analytics amongst collaborators.

Observational Medical Outcomes Partnership (OMOP)

A public-private partnership established in the US to inform the appropriate use of observational healthcare databases for studying the effects of medical products. The five-year project developed new methods in observational research and established an observational research laboratory.

A centerpiece of the OMOP project was the development of the OMOP Common Data Model. While the project ended, the name for the common data model stuck; the whole ecosystem is now developed by the OHDSI and collaborators around the world.

OMOP Common Data Model (OMOP CDM)

A data model which can represent healthcare data from diverse sources in a consistent and standardized way. Is a "strong" information model, in which the encoding and relationships among concepts are explicitly and formally specified. *See also*: OMOP, OHDSI.

Reference Information Model (RIM)

Single information model that covers the domain of activity being addressed by a standards developing organization using this methodology.

In the context of this work, the data model for HL7 Version 3. The RIM describes the kinds of information that may be transmitted within health-care organizations, and includes *acts* that may take place (procedures, observations, interventions, and so on), relationships among acts, the manner in which health-care personnel, patients, and other entities may participate in such acts, and the roles that can be assumed by the participants (patient, provider, specimen, and so on).

refset

An abbreviation for SNOMED CT reference set, a standard format for maintaining and distributing a set of references to SNOMED CT components.

Regenstrief LOINC Mapping Assistant (RELMA)

A Windows-based mapping utility for searching the LOINC database and mapping local codes to LOINC codes, provided by Regenstrief Institute.

The RELMA software has been placed in a maintenance mode and will be eventually discontinued. For now, RELMA will continue to be updated with each LOINC release but no new features will be added. Instead, its functions such as viewing hierarchies and term mapping will be migrated to web-based applications.

role

A concept used in PostgreSQL to manage manages database access permissions.

A role can be thought of as either a database user, or a group of database users, depending on how the role is set up. Roles can own database objects (for example, tables and functions) and can assign privileges on those objects to other roles to control who has access to which objects. Furthermore, it is possible to grant *membership* in a role to another role, thus allowing the member role to use privileges assigned to another role.

RxNorm

Standardized nomenclature for clinical drugs. The name of a drug combines its ingredients, strengths, and/or form. Links to many of the drug vocabularies commonly used in pharmacy management and drug interaction software.

Developed by the National Library of Medicine (US).

schema

In a PostgreSQL database system, a namespace for SQL objects, which all reside in the same database. Each SQL object must reside in exactly one schema. More generically, the term *schema* is used to mean all data descriptions (table definitions, constraints, comments, etc) for a given database or subset thereof.

secondary use of data

Non-direct care use of personal health information including but not limited to analysis, research, quality/safety measurement, public health.

SNOMED CT Identifier (SCTID)

A unique integer identifier applied to each SNOMED CT component (Concept, Description, Relationship); is between 6 and 18 digits long. This data type is used to identify SNOMED components, to refer to a component from another component or from a SNOMED CT reference set, and also to represent the values for concept enumerations.

An example would be `428041000124106` to identify a concept with term `Occasional tobacco smoker (finding)`.

SNOMED CT reference set

A standard format for maintaining and distributing a set of references to SNOMED CT components.

- A reference set can be used to represent a subset of components (concepts, descriptions or relationships).
- A reference set may also associate referenced components with additional information such as:
  - Ordered lists of components
  - Sets of associations between components
  - Mapping between SNOMED CT concepts and other systems codes, classifications, or knowledge resources.

Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)

A comprehensive, multilingual clinical healthcare terminology that can be processed electronically; useful for enhancing the standardized use of medical terms in clinical systems.

Owned, administered and developed by the International Health Terminology Standards Development Organisation (IHTSDO), aka SNOMED International (UK).

terminology

A set of terms representing the system of concepts of a particular subject field.

II. Top-level Section Types in CDA Documents

Table 1. Top-level section types in CDA documents (using EGCUT dataset).

		Amount		
code	**Various `displayName` attribute values**	**Count**	**Percent (%)**	**25%**
AMBS	Ambulatoorne haigusjuht Ambulatoorne haigusjuhtum	6 423 821	21.146	
DGN	Diagnoos Loplik kliiniline diagnoos L?plik kliiniline diagnoos Lõplik kliiniline diagnoos L&#65533;plik kliiniline diagnoos	5 169 584	17.017	
SUM	Kokkuvote patsiendi ravist Kokkuv?te patsiendi ravist Kokkuvõte patsiendi ravist Kokkuv&#65533;te patsiendi ravist	4 634 002	15.254	
ANAM	Anamnees Anamnees, diagnoosi p?hjendus ja haiguse kulg Anamnees, diagnoosi pohjendus ja haiguse kulg Anamnees, diagnoosi põhjendus ja haiguse kulg Anamnees, diagnoosi p&#65533;hjendus ja haiguse kulg Anamnees, kaebus Objektiivne leid	4 486 867	14.77	
OBJFIND	Objektiivne leid	2 466 549	8.119	
DRUG	Valjastatud ravimid V?ljastatud ravimid Väljastatud ravimid V&#65533;ljastatud ravimid	1 765 431	5.811	
PROC	Uuringud ja protseduurid	1 266 023	4.167	

Table 1. (continued; codes with less than 0.5% not shown)

code	Various displayName attribute values	Amount		
		Count	Percent (%)	25%
STATE	Seisund väljakirjutamisel Seisund v&#65533;ljakirjutamisel	1 189 173	3.914	
ANA	Anal??sid Analüüsid Anal&#65533;&#65533;sid Laboratoorsed uuringud Teostatud analüüsid Uuringud ja protseduurid	986 118	3.246	
REGIME	Re?iimi ja ravialased soovitused Re¸iimi ja ravialased soovitused Reziimi ja ravialased soovitused Režiimi ja ravialased soovitused Re&#65533;iimi ja ravialased soovitused	887 298	2.921	
DOC	V?ljastatud dokumendid Väljastatud dokumendid V&#65533;ljastatud dokumendid	309 796	1.02	
HOSP	Haiglas viibimine	292 655	0.963	

III. Licence

Non-exclusive licence to reproduce the thesis and make the thesis public

I, **Harry-Anton Talvik**,
(author's name)

1. grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

   Workflow for Transforming Health Records to OMOP Common Data Model,
   (title of thesis)

   supervised by Sven Laur, Raivo Kolde and Sulev Reisberg.
   (supervisor's name)

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY-NC-ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Harry-Anton Talvik
17/05/2022