HOCHSCHULE HEILBRONN
HEILBRONN UNIVERSITY
ENGINEERING BUSINESS INFORMATICS

Heidelberg University
Heilbronn University

Integration of i2b2 into the Greifswald University Hospital Research Platform

Master's Thesis

submitted by

Tobias Bronsch

in fulfillment of the requirements for the degree of

Master of Science (M.Sc.)

January 2014

Primary thesis supervisor:	Prof. Dr. med. Björn Bergh, Heidelberg University
Secondary thesis supervisor:	Prof. Dr. med. Wolfgang Hoffmann, MPH, Greifswald University

Affidavit

I hereby declare that the following master's thesis "*Integration of i2b2 into the Greifswald University Hospital Research Platform*" has been written only by the undersigned and without any assistance from third persons.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Place, Date

Signature

Matriculation numbers:

179670	(Heilbronn University)
3076667	(Heidelberg University)

To my love, 肖榕.

李白《夜思》

床前明月光，
疑是地上霜。
望明月，
低思故。

Thoughts in the Silent Night
(by *Li Bai*)

Beside my bed a pool of light -
Is it hoarfrost on the ground?
I lift my eyes and see the moon,
I bend my head and think of home.

Acknowledgements

I would like to thank Prof. Bergh and Prof. Hoffmann for accepting my thesis. Special thanks goes to Dr. Ulrike Kutscha, Robert Gött, Lars Geidel, Theresia Eichner and Christoph Havemann, who acted as a great support in giving me feedback for my work. I also thank all the proof readers of my thesis.

Abstract

The Greifswald University Hospital in Germany conducts a research project called *Greifswald Approach to Individualized Medicine* (GANI_MED), which aims at improving patient care through personalized medicine. As a result of this project, there are multiple regional patient cohorts set up for different common diseases. The collected data of these cohorts will act as a resource for epidemiological research.

Researchers are going to get the possibility to use this data for their study, by utilizing a variety of different descriptive metadata attributes. The actual medical datasets of the patients are integrated from multiple clinical information systems and medical devices. Yet, at this point in the process of defining a research query, researchers do not have proper tools to query for existing patient data. There are no tools available which offer a metadata catalogue that is linked to observational data, which would allow convenient research. Instead, researchers have to issue an application for selected variables that fit the conditions of their study, and wait for the results. That leaves the researchers not knowing in advance, whether there are enough (or any) patients fitting the specified inclusion and exclusion criteria.

The *Informatics for Integrating Biology and the Bedside* (i2b2) framework has been assessed and implemented as a prototypical evaluation instance for solving this issue. i2b2 will be set up at the *Institute for Community Medicine* (ICM) at Greifswald, in order to act as a preliminary query tool for researchers.

As a result, the development of a research data import routine and customizations of the i2b2 webclient were successfully performed. An important part of the solution is, that the metadata import can adapt to changes in the metadata. New metadata items can be added without changing the import program. The results of this work are discussed and a further outlook is described in this thesis.

Thesis overview

1 Introduction 13

1.1 Project GANI_MED 15

1.2 Problem 16

1.3 Thesis goals 17

2 Fundamentals 20

2.1 Data warehouse 20

2.2 Metadata 21

2.3 EAV model 22

2.4 Query definition at the transfer unit 23

3 Methods 25

3.1 Environment 25

3.2 Applied tools 32

3.3 Approach 36

4 Results 40

4.1 First goal: Integration of metadata and research data into i2b2 40

4.2 Second goal: Integration of the i2b2 webclient query 58

4.3 Example query 65

5 Discussion 70

5.1 Thesis goals 70

5.2 Effort and complexity of the solution 70

5.3 Maintainability 72

5.4 Suitability of the applied tools 73

5.5 Independence from changing metadata or observational data . 73

5.6 Secondary data warehouse 74

5.7 User management in i2b2 77

5.8 Data security measures and concerns 77

5.9 Alternative to a customized webclient 80

5.10 Further i2b2 webclient development 81

5.11 Related works 81

6 Outlook **83**
6.1 Key user evaluation of i2b2 83
6.2 Import of the researcher's query into the transfer unit's web application 83
6.3 Data privacy 83

A Appendix **86**
A.1 Screenshots 86
A.2 ETL process for loading the research database 87
A.3 Process model of the research data application process 88
A.4 BPMN model of the research data application process 89
A.5 G2D2 DTO models 90
A.6 XML: Example export query from i2b2 webclient 92
A.7 XML: Swisslab G2D2 metadata, i2b2 related AttributeTypeDefs 95

Acronyms **96**

List of Figures **99**

List of Tables **100**

References **101**

Table of contents

1 Introduction 13

1.1 Project GANI_MED 15

1.2 Problem 16

1.2.1 Research platform and the *Transferstelle für Daten- und Biomaterialienmanagement* at Greifswald University Hospital 16

1.2.2 Problem description 16

1.3 Thesis goals 17

1.3.1 First goal: Integration of metadata and research data into i2b2 18

1.3.2 Second goal: Integration of the i2b2 webclient query . . . 18

2 Fundamentals 20

2.1 Data warehouse 20

2.1.1 Data cleansing 20

2.1.2 Primary and secondary data warehouse 20

2.1.2.1 Primary data warehouse 20

2.1.2.2 Secondary data warehouse 21

2.2 Metadata 21

2.3 EAV model 22

2.4 Query definition at the transfer unit 23

3 Methods 25

3.1 Environment 25

3.1.1 Research platform 25

3.1.2 Primary data warehouse 25

3.1.2.1 Database schema 27

3.1.3 Development of the web application at the transfer unit . 29

3.1.4 BPMN process of the data application at the transfer unit 30

3.1.5 The generic metadata repository G2D2 31

3.1.5.1 General explanations 31

3.2 Applied tools 32

3.2.1 Software technology 32

3.2.1.1 Eclipse 32

3.2.1.2 Aptana Studio 3 32

3.2.1.3 Java . . . 32
3.2.1.4 JavaScript . . . 33
3.2.1.5 FileSaver.js . . . 33
3.2.1.6 Oracle XE . . . 33
3.2.1.7 Ubuntu operating system . . . 33
3.2.1.8 UML . . . 34
3.2.1.9 i2b2 . . . 34
3.2.1.10 i2b2 Wizard . . . 35
3.3 Approach . . . 36
3.3.1 Metadata import . . . 36
3.3.1.1 Software architecture . . . 37
3.3.2 Planned i2b2 webclient customization . . . 39

4 Results **40**
4.1 First goal: Integration of metadata and research data into i2b2 40
4.1.1 Importing research data into the i2b2 star schema . . . 48
4.1.1.1 Data cleansing and quality assurance . . . 54
4.1.2 Data cleansing logs . . . 55
4.1.3 Import performance comparison . . . 56
4.2 Second goal: Integration of the i2b2 webclient query . . . 58
4.2.1 Requirements of query exchange of i2b2 with the transfer unit . . . 58
4.2.2 Planned i2b2 webclient customization . . . 59
4.2.3 Implemented customization of the i2b2 webclient . . . 60
4.2.4 Implementation of the *Choose variables* tab . . . 60
4.2.5 *Choose variables* tab data model . . . 62
4.2.6 Data export functionality . . . 63
4.2.7 Query tool metadata export . . . 64
4.2.8 *Choose variables* metadata export . . . 64
4.3 Example query . . . 65
4.3.1 Defining inclusion and exclusion criteria . . . 65
4.3.2 Set up query for the transfer unit . . . 67

5 Discussion **70**
5.1 Thesis goals . . . 70

5.1.1 First goal: Integration of metadata and research data into i2b2 . . . 70
5.1.2 Second goal: Integration of the i2b2 webclient query . . . 70
5.2 Effort and complexity of the solution . . . 70
5.2.1 Data import into i2b2 . . . 71
5.2.2 Customized i2b2 webclient . . . 71
5.3 Maintainability . . . 72
5.4 Suitability of the applied tools . . . 73
5.5 Independence from changing metadata or observational data . 73
5.6 Secondary data warehouse . . . 74
5.6.1 Reasons for using a secondary data warehouse . . . 74
5.6.2 Attributes . . . 74
5.6.3 Objective . . . 75
5.6.4 Requirements . . . 76
5.6.5 Reasons for using i2b2 for this scenario . . . 76
5.7 User management in i2b2 . . . 77
5.8 Data security measures and concerns . . . 77
5.8.1 Data privacy . . . 77
5.8.1.1 i2b2 patient count functionality . . . 78
5.8.1.2 i2b2 metadata tree at Greifswald . . . 79
5.8.2 File generation for query export . . . 80
5.9 Alternative to a customized webclient . . . 80
5.10 Further i2b2 webclient development . . . 81
5.11 Related works . . . 81
5.11.1 IDRT - Integrated Data Repository Toolkit . . . 81
5.11.2 HL7 ETL cell . . . 82

6 Outlook 83
6.1 Key user evaluation of i2b2 . . . 83
6.2 Import of the researcher's query into the transfer unit's web application . . . 83
6.3 Data privacy . . . 83

A Appendix 86
A.1 Screenshots . . . 86
A.2 ETL process for loading the research database . . . 87

A.3 Process model of the research data application process 88
A.4 BPMN model of the research data application process 89
A.5 G2D2 DTO models 90
A.6 XML: Example export query from i2b2 webclient 92
A.7 XML: Swisslab G2D2 metadata, i2b2 related AttributeTypeDefs 95

Acronyms 96

List of Figures 99

List of Tables 100

References 101

1 Introduction

The value of collecting and reusing clinical data for research is increasingly recognized. A strong evidence for this development is the increased usage of terms or keywords in research publications, which relate to the reuse of clinical data. In recent years, terms like *Big data*, *data warehouse* and *secondary use* can be more frequently found in medical research publications (Figure 1, [1]).

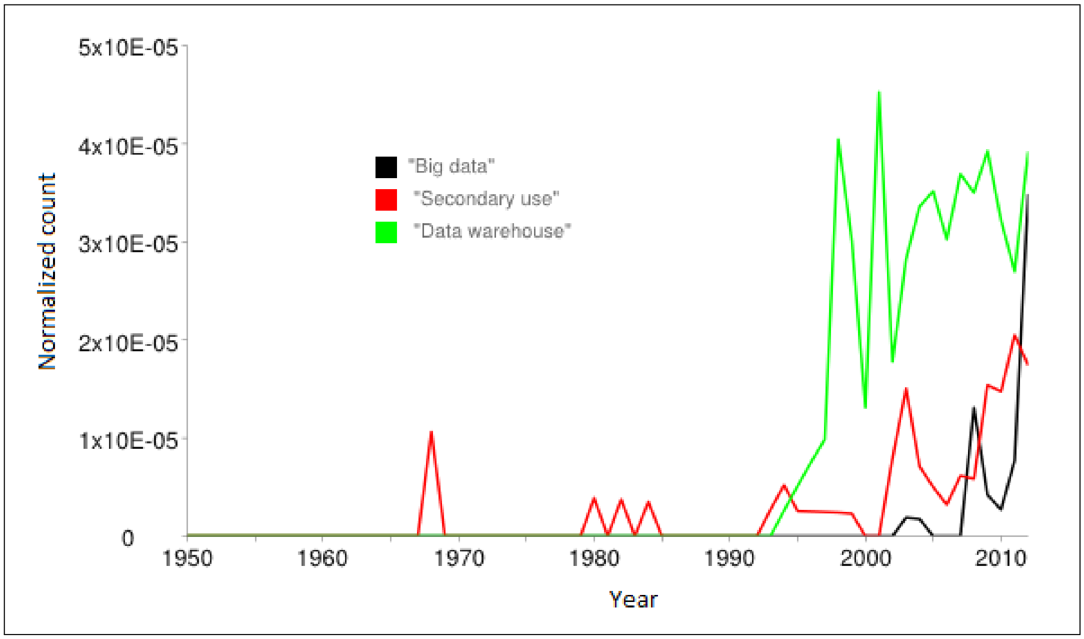


Figure 1: Term frequency of terms related to the reuse of clinical data on Medline, using MLTrends search engine, by searching in publication's titles and abstracts, normalized over publication count [1].

A patient's data is today more often stored in *Electronic Health Records* (EHRs) [2]. That data is mainly meant to be used in the treatment context [3]. Yet, beside the treatment context, there are also research-related secondary uses, e.g.:

- ”Disease specific clinical or epidemiological research projects
- Health care research, assessment of treatment quality, health economy” ([3], page 1, Introduction)

The data contained in an EHR may come from a variety of sources. This could e.g. be different clinical or laboratory departments of a hospital. The secondary use of this data may face quality issues, due to the integration of the different source data [4]. Yet, the integrated data in EHRs makes it a very valuable data source for research. It allows to view patient data on an interdisciplinary scale that may not be possible otherwise. This may allow a holistic view on the patient's medical conditions and history.

Another interesting development is the increased ongoing research in the fields of individualized or personalized medicine and biological data (Figure 2, [1]).

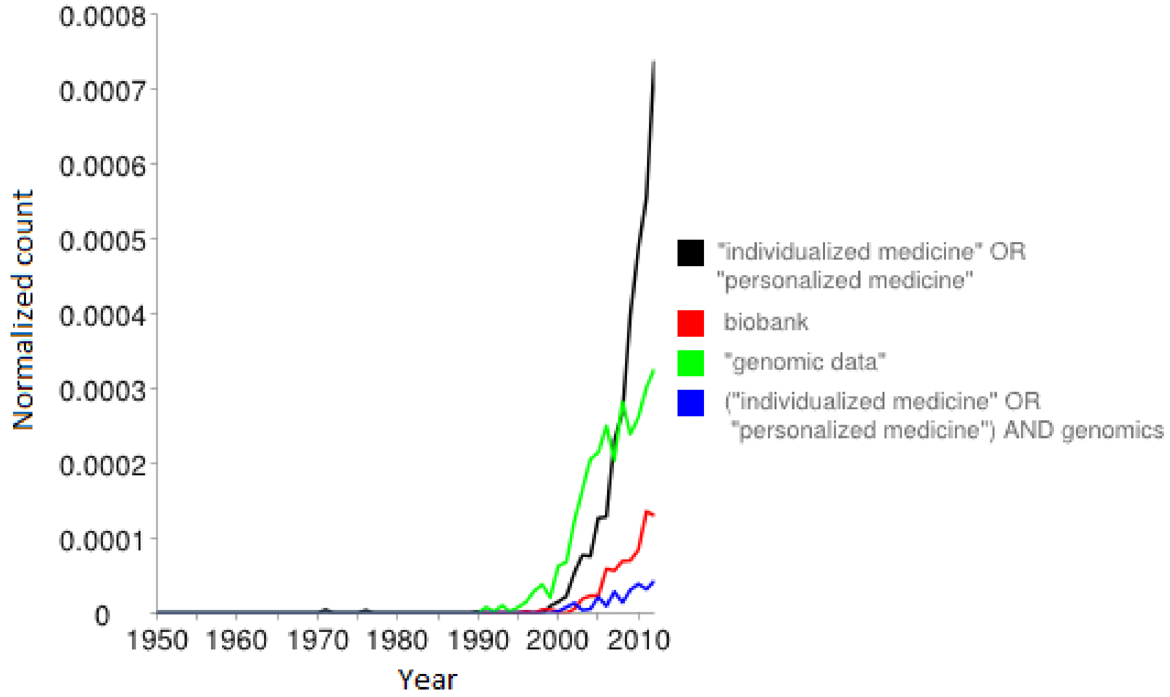


Figure 2: Term frequency of terms related to individualized medical treatment and biological data on Medline, using MLTrends search engine, by searching in publication's titles and abstracts, normalized over publication count [1].

The classical approach of understanding diseases aims at their pathology and at the treatment of their signs and symptoms. Personalized medicine is supposed to replace the classical approach. Rather, underlying biological mechanisms should be modelled and *'-omics'* data (e.g. *genomics*, *proteomics*, *metabolomics*) should be used instead to understand and treat diseases in the future [5].

Responses of an individual to drugs are found to be related to a patient's genome or other molecular mechanisms. It is believed that individually customized therapies can improve patient care [6].

Another promised effect of personalized medicine is the reduction of treatment costs in some cases [7]. Yet, with the advance of the *'-omics'* technologies, unresolved microeconomic problems occur, limiting the possibilities of personalized medicine [8]. For the patient there may be a better outcome despite economic problems.

Many promises of an improved health care come with the advent of modern biotechnology, the ability of storing vast amounts of information about patients (*Big data*) and the idea of reusing patient records for research (*secondary use*).

One ongoing research project that has set its goal to achieve great improvements for health care is the GANI_MED project conducted at Greifswald University Hospital, Germany. This project will be introduced in the following sections.

1.1 Project GANI_MED

The project *Greifswald Approach to Individualized Medicine* (GANI_MED) comprises multiple national and international institutions and industrial partners. The project's goal is to study personalized medicine on an interdisciplinary scale, developing new analytical procedures for various diseases that are present in the general population. These diseases include heart disease, stroke, renal failure, and others. The GANI_MED-consortium provides for this purpose the infrastructure for biobanking, bioinformatics and medical informatics which are a key part of the research project. The GANI_MED project is considered to identify promising medical treatment concepts which are most suitable to individual patients [9].

The *Institute for Community Medicine* (ICM) at Greifswald University Hospital is a key part of the project. The institute conducts - amongst others - epidemiological and methods research in the field of community medicine, and develops the necessary informatics infrastructures for the GANI_MED project [10]. Furthermore, the ICM can refer to many years of experience with epidemiological studies [11].

Extensive measures are taken in order to meet the standards of related laws and data privacy acts, as well as ethical standards. In contrast to other projects, there needs to be an *opt-in* mechanism to ensure a patient's consent to the use of his or her personal and medical data as well as biomaterial [12].

An important part of the project is the research platform. The data in the research platform is organized as a data warehouse which integrates the data from multiple sources.

1.2 Problem

1.2.1 Research platform and the *Transferstelle für Daten- und Biomaterialienmanagement* at Greifswald University Hospital

The data which the research platform comprises are integrated from clinical documentation, basic documentation of the epidemiological cohorts and medical devices. That makes it convenient for the researcher to get a holistic view on the existing medical data, allowing him to conduct interdisciplinary research.

In order to transfer the stored data and biomaterial to researchers, there is an administrative unit established, called *Transferstelle für Daten- und Biomaterialienmanagement* (eng.: *Management and Transfer of Data and Biomaterials*). Researchers can request medical data about patients, medical images or biomaterial there. In order to facilitate the application process, there is a web application in development where researchers are able to submit requests for epidemiological data.

After submitting the query to the transfer unit of the research platform, the query is reviewed and the resultset is manually generated by the personnel at the transfer unit. This means that the researcher doesn't get an immediate response.

1.2.2 Problem description

The research platform itself is a useful tool for researchers to acquire detailed datasets for their analysis. Yet, one problem with this approach arises, which is, that there is no possibility to tell the researcher at the time of his query, how many patients are actually existing in the database that fit his inclusion and exclusion criteria. In other words, the problem of the current situation is, that the results that a researcher receives come with a time delay.

The researchers have to rely on their experience telling them, that they may just have enough patients of a certain kind. These experiences could come e.g. from clinical work with such patients. The result could then be, that a researcher is confident to find enough patients for his study, but ends up with a too small dataset. In extreme cases, there may be no data at all for the specified query.

To overcome this problem, there needs to be a way to tell the researcher immediately, how many patients fit the specified criteria. This should be done before the query is submitted to the transfer unit. This approach allows the researcher to figure out, whether he will have a resultset that is reasonably large enough in size for his study.

Additionally, the researcher could have a more direct access to the underlying research database. This allows him to ask many different questions at the same time, before actually submitting his query to the transfer unit. The ability of querying the research platform data in advance allows evaluating new research topics without waiting for the output for each of a researcher's ideas.

There is an ongoing research project called *i2b2* (abbr. *Informatics for Integrating Biology and the Bedside*), which can be used for this exact purpose and which is more and more widely used. As a summary, this master's thesis aims at answering the question, whether i2b2 can be integrated into the research platform at Greifswald. In other words, whether the gap between researchers and the research platform itself can be bridged by i2b2.

1.3 Thesis goals

This thesis consists of two main technical goals that are discussed individually in the following sections.

The first goal is the integration of GANI_MED specific metadata called *Generic GANI_MED Data Dictionary* (G2D2) with actual observational data - e.g. laboratory results - into i2b2. The observational data is referred to as the research platform database. It should be possible, to share the G2D2 metadata references between i2b2 and the transfer unit. This should allow a transfer of the researcher's query from i2b2 into the transfer unit.

The second goal is the integration of the i2b2 query with the transfer unit via a standardized exchange format, containing the selected metadata items.

1.3.1 First goal: Integration of metadata and research data into i2b2

For the first goal an import tool needs to be developed. This tool is supposed to be able to import both, the G2D2 metadata and the observational data into the i2b2 database.

It is considered useful to write a generic import program in such a way, so that the program code doesn't have to be altered in the future. Changes in the metadata or observational data should not affect the import program code. This approach should keep the maintenance work low. Due to the fact that both i2b2 and the research database are designed in an *Entity-Attribute-Value* (EAV) model, the maintenance work remains very low if there are changes happening to them in the future.

Furthermore, the import of observational data should be configurable in such a way, that it allows to skip observational data imports for certain metadata, if necessary.

The program should be able to integrate G2D2 metadata with research platform data reliably. It should be dependent on the validity of the underlying data only, not on specific datasets or items. The data import can then be controlled through changing the metadata (e.g. adding new items) and observational data (e.g. by storing more patient data). Yet, the program code doesn't have to be altered anymore.

The import tool should also be running efficiently to possibly import millions of datasets in a reasonable time. It is expected that this importing process will not be running every day, but rather every quarter of a year, after a new quality assured research database release was generated. Yet, the program should still run with reasonable performance and possible poor performance should be tracked down through software optimization to keep hardware requirements at a minimum.

1.3.2 Second goal: Integration of the i2b2 webclient query

The second thesis goal is the integration of the researcher's query from the i2b2 webclient into the web application of the transfer unit. As a first step, the i2b2 query will be forwarded to the transfer unit through a file based approach by a shared, standardized file format. The researcher will have to store his query in a local file being downloaded from the i2b2 webclient. That file needs to be uploaded later to the transfer unit's web application, in order to set the selected i2b2 items automatically in the transfer unit's web application.

The ability to automatically set the selected metadata from i2b2 in the transfer unit's web application is a crucial part of the whole process. Beside additional metadata, the transfer unit contains the same metadata as i2b2. The export functionality of the selected metadata from i2b2 is a part of this master's thesis (Section 4.2.6). The import of that query into the transfer unit's web application will be developed at Greifswald.

In order to give the researcher the ability to make a detailed query, the i2b2 webclient is being customized in this thesis, because the necessary functionality is not there yet. The existing query tool within the i2b2 webclient was developed to query the database for a few main criteria. These are called the inclusion and exclusion criteria for the study. If the researcher finds out, that there are enough patients in the research database fitting his criteria, he then can select detailed variables in the customized webclient. These selected, detailed variables will later be used by the transfer unit to generate the detailed epidemiological output for the researcher.

The additional, detailed variable selection is splitted into three parts, namely *Exposure*, *Outcome* and *Other* variables. This allows to inform the transfer unit about the intention of the researcher, indicating in which context a variable is to be used. An example would be to query for *smoking* as an exposure and for *lung cancer* as an outcome. A researcher would submit that query to the transfer unit in order to apply for a detailed data output.

By using the tool developed in the scope of this thesis, i2b2 provides the researcher only with an absolute patient count, depending on what inclusion and exclusion criteria are specified.

2 Fundamentals

This section explains fundamental concepts that this thesis is based on.

2.1 Data warehouse

A data warehouse in general is an information system that can be loaded by integrating data from multiple sources and which offers users the possibility of querying that integrated data ([13], Page 5, *Abgrenzung und Einordnung*).

2.1.1 Data cleansing

The integration of source data creates the need for extensive data cleansing procedures during the data import [14]. Because the data is integrated from multiple sources, the loading of the data warehouse takes extensive transformation processes to merge and clean data before it is stored in the data warehouse ([13], Page 101, *3.3.2 Bereinigung*).

This data cleansing is also necessary, because source systems can contain data that has no quality assurance. An example of non-clean data may be a laboratory value in a pending state. That kind of data is not meant to be queried by a researcher, therefore it has to be cleaned before the import.

The process of skipping non desired data from the data import is referred to as *data cleansing*.

2.1.2 Primary and secondary data warehouse

In this thesis, there is a distinction between a *primary* and a *secondary* data warehouse. The meaning of these two is explained below.

2.1.2.1 Primary data warehouse

The *primary* data warehouse is a database system that contains integrated data from multiple sources, such as clinical information systems. This data warehouse acts as a basis for other data warehouses that can be derived from the primary data warehouse.

The primary data warehouse is loaded through an *Extract, transform, load* (ETL) process. This process extracts raw data from productive information systems, transforms the data and loads them into the primary data warehouse.

2.1.2.2 Secondary data warehouse

The *secondary* data warehouse is a system that is loaded using data from the primary data warehouse and offers users the possibility of querying that data. The imported data already went through quality assurance in the primary data warehouse. Yet, in order to fit the secondary data warehouse's database schema and application requirements, the data may need another data cleansing step.

An example may be that a certain metadata item in the secondary data warehouse defines a variable to be a numerical variable, but the primary data warehouse contains instead a list of textual items for that variable. This data should not be imported into the secondary data warehouse. Filtering such values may be achieved by using whitelist checks and other rules during the import routine. Skipped metadata items may be stored in log files in order to inform the user about the filter process.

2.2 Metadata

Metadata in general means *data about data* [15]. Data may be divided into the actual data, e.g. the text of a book, and the data describing that book, e.g. the title. The fact, that a book *has* e.g. a title, is metadata. The *actual text* of the title is again part of the *actual data* of the book. This approach allows e.g. to create indexes or data dictionaries, which allow for searching for book titles or other items of interest.

In this thesis there is a divide between data that is considered a catalogue of e.g. medical terms that a researcher is interested in (metadata), and the data that is coming from e.g. a clinical information system (observational data).

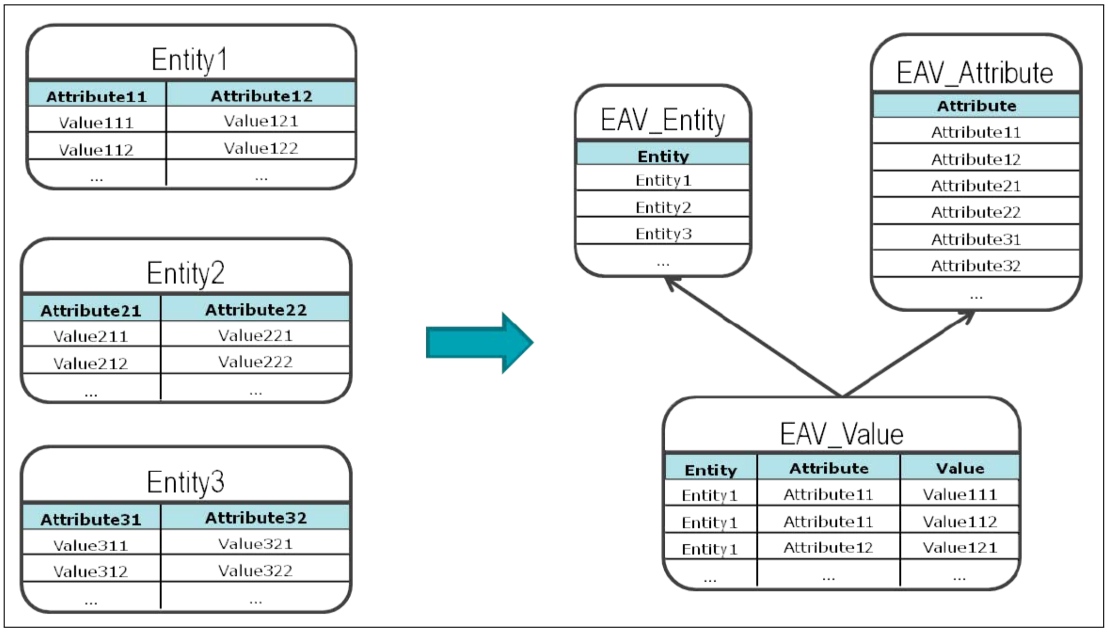
The metadata here is a relatively large catalogue of a hierarchical nature, that contains all the items that a researcher could be interested in, but it doesn't contain actual observational data.

An example could be the laboratory value ”hemoglobin”. This item can be considered a metadata item that may have parent items such as ”blood count”, which in turn may be part of ”hematology”. The actual observational hemoglobin values are yet unknown in the metadata.

The metadata catalogue needs to be linked to the actual existing research data from the primary data warehouse. For this purpose, there needs to be a data importing procedure to read, link and store the metadata and research data together in the secondary data warehouse, which can then be queried by a researcher.

2.3 EAV model

The EAV model is a database paradigm that divides a database into relatively few tables, referred to as the entity, attribute and value tables. The entity table may contain a reference to an observation, whereas the attribute table may contain e.g. a laboratory variable. In the value table, the actual values related to the attributes and entities are stored. This means that the attribute table contains metadata, whereas the value table contains the actual values of the entity and the attribute table. The entity and attribute table are linked to each other through the value table (Figure 3, [16]).



Entity1

Attribute11	Attribute12
Value111	Value121
Value112	Value122
...	...

Entity2

Attribute21	Attribute22
Value211	Value221
Value212	Value222
...	...

Entity3

Attribute31	Attribute32
Value311	Value321
Value312	Value322
...	...

EAV_Entity

Entity
Entity1
Entity2
Entity3
...

EAV_Attribute

Attribute
Attribute11
Attribute12
Attribute21
Attribute22
Attribute31
Attribute32
...

EAV_Value

Entity	Attribute	Value
Entity1	Attribute11	Value111
Entity1	Attribute11	Value112
Entity1	Attribute12	Value121
...	...	...

Figure 3: Entity-attribute-value format compared with the classical relational approach [16].

This approach allows to hide the complexity of the contained data in the table rows instead of the columns. That means that it is rather simple to add to or remove an arbitrary amount of entities, attributes or values from the database.

In classical relational databases, using tables with many columns, there would be the need to manually alter the database schema instead. A new metadata item, like a laboratory variable, may require a new column in a laboratory table. In the clinical context, there would be the need to alter the database continuously. This is due additional information systems or other data sources that need to be included.

Also, the potentially huge amount of laboratory items would require many tables. The reasons for that are *Relational Database Management System* (RDBMS) restrictions, which allow only a limited amount of columns per table (e.g. [17]).

The advantages of an EAV formatted database yet come with a major downside, that is that analytical processing is considered unsuitable for that kind of databases [18].

2.4 Query definition at the transfer unit

The web application at the transfer unit allows the specification of variables as being an *Exposure*, an *Outcome* or a *Chosen* variable. The meaning of that categories is explained below (Table 1):

Item group	Meaning
Exposure	All items, which a patient is exposed to (e.g. smoking)
Outcome	All outcome items (e.g. heart disease)
Chosen	All other items, which are neither exposure, nor outcome. This can contain confounders.

Table 1: Additional G2D2 attributes for i2b2 ontology items.

A probable scenario is the following: A researcher is interested in the medical status of patients that are smoking (*exposure* items), who developed a coronary heart disease (*outcome* item) and where certain laboratory values, like the complete blood count are selected as the *chosen* items. The laboratory values are neither an exposure, nor an outcome. The transfer unit now has to interpret this specified query and return the resulting patient datasets to the researcher.

The interpretation of which variable is a confounding variable is up to the team at the transfer unit that generates the output. A variable that is chosen as *exposure* or *outcome* is automatically selected as a *chosen* item.

3 Methods

In this section, the methods of this work are described. This section is divided into an *Environment*, an *Applied tools* and an *Approach* section.

3.1 Environment

This section describes the found external requirements which are related to this thesis. These environmental requirements are considered as being fixed and are introduced below.

3.1.1 Research platform

The research platform at Greifswald acts as a means of integrating and storing research data and is a core part of the GANI_MED project [19]. The research platform contains its own ETL process. In this ETL process there are different interfaces to services such as pseudonymization or a master patient index. In the following sections, the primary data warehouse contained in the research platform is introduced.

3.1.2 Primary data warehouse

The primary data warehouse is loaded by executing an ETL process that extracts data from source information systems and integrates them into the research database. The integrated data comprises the general patient documentation as well as lab values, dental and electrocardiography data. During the process, there are extensive transformation and pseudonymization processes that aim at guaranteeing data privacy of the patients. A detailed view of the ETL process that is used to load the primary data warehouse can be found in the appendix (Figure 23).

Due to the fact that the primary data warehouse contains data from multiple clinical source information systems, the data is potentially unclean. An example for that would be laboratory values which are in a pending state, or values that were manually entered. These unclean values are not meant to be of any use to researchers and should be kept out of the secondary data warehouse.

Additionally, the primary data warehouse is compiled into a release version every quarter of a year. That release version of the primary data warehouse is quality assured, the data has been cleaned and it acts as the basis for the secondary data warehouse.

Below is an overview of the relationship beween the ETL process, the primary and secondary data warehouse, the quality assurance and a researcher (Figure 4, [20]).

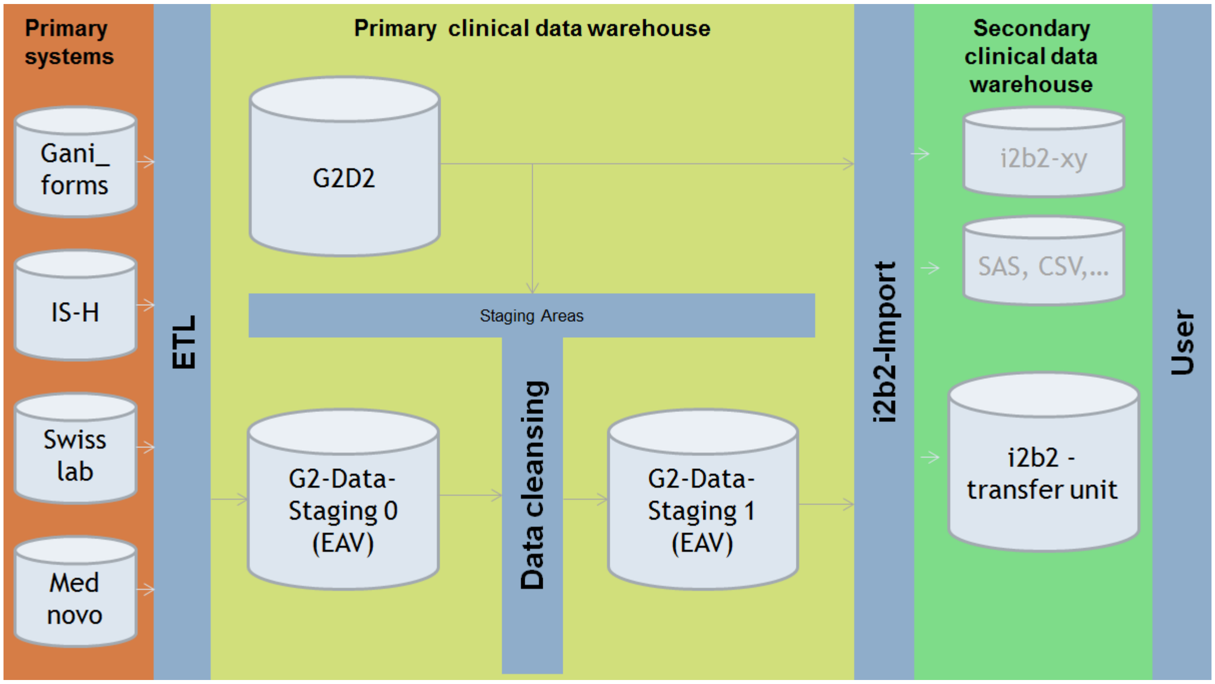


Figure 4: Systematic overview of the ETL process, the primary / secondary data warehouse and the quality assurance process [20].

On the left of the above image are the primary information systems that are used by the ETL process to load the primary data warehouse. The primary information systems currently are:

- GANI_forms
- IS-H
- Swisslab
- Medical devices

The GANI_forms information system stores data about patients who answered the GANI_forms questionnaire. This questionnaire is developed as a part of the GANI_MED project.

IS-H stores the basic data about patients, as well as how patients are admitted to or discharged from different hospital departments.

Swisslab is the laboratory information system and stores lab values of different patient's specimen.

Another information source are medical devices, which are integrated using a common information interface. Currently, it is possible to include data from dentistry and from long-term *Electrocardiography* (ECG) measurements.

In the research platform, there are two staging areas. The first staging area is the database system that is loaded by the ETL process. *Stage 0* is then going through quality assurance at Greifswald and is matched to G2D2. The *stage 1* database and G2D2 act as the input for the import routine to the secondary data warehouse. On the right, the user is indicated. That means that this is the place where an application is located, which can query the secondary data warehouse.

The i2b2 import routine may also be used to create different instances of secondary data warehouses, creating a variety of different i2b2 instances (Section 4.1.1).

3.1.2.1 Database schema

The internal structure of the primary data warehouse in general is organized into *entities*, *obervations* and *values*. As entities, there may be patients as well as physicians, or even information systems. Each entity has its own unique identifier. If the entity is for instance a patient, then that identifier is the patient identifier.

The *entities* and *values* are connected through *observations*. The actual values of these observations and entities are stored by creating a stack of these values. This approach was chosen in order to provide a generic structure of the data warehouse. That allows the data warehouse to support a variety of study types and study designs without the need of changing the database design.

The before mentioned entities and values in this high level view should not be misunderstood as being entities and values of an EAV model. Below are two models in order to help understand this generic approach (Figures

5 and 6, [21]).

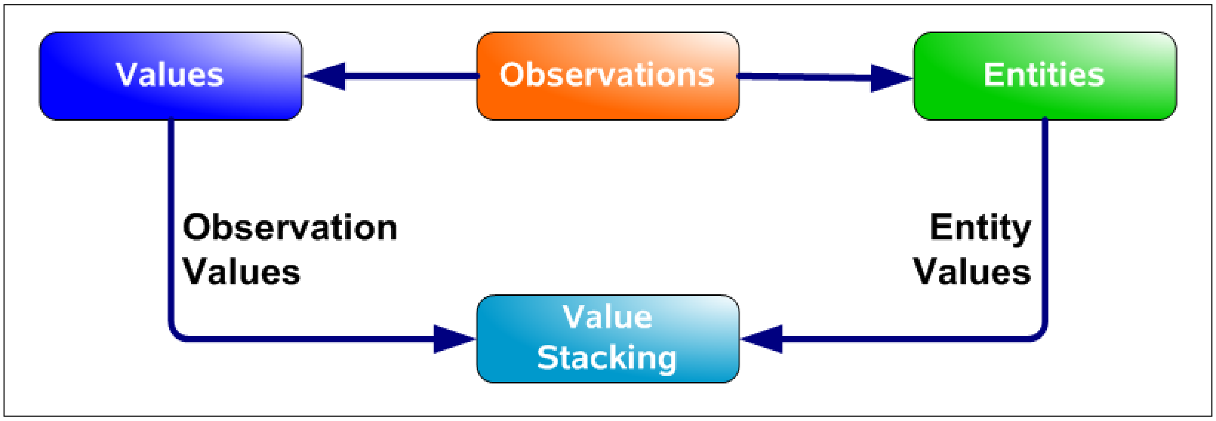


Figure 5: General *high level* structure of the values, observations and entities of the research database [21].

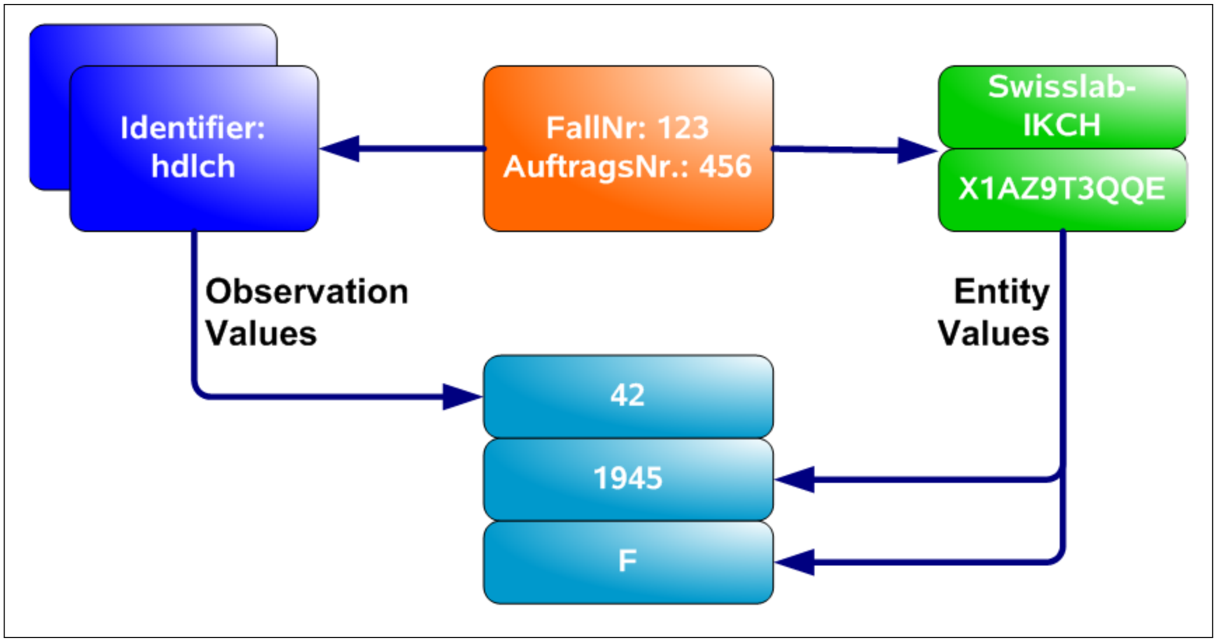


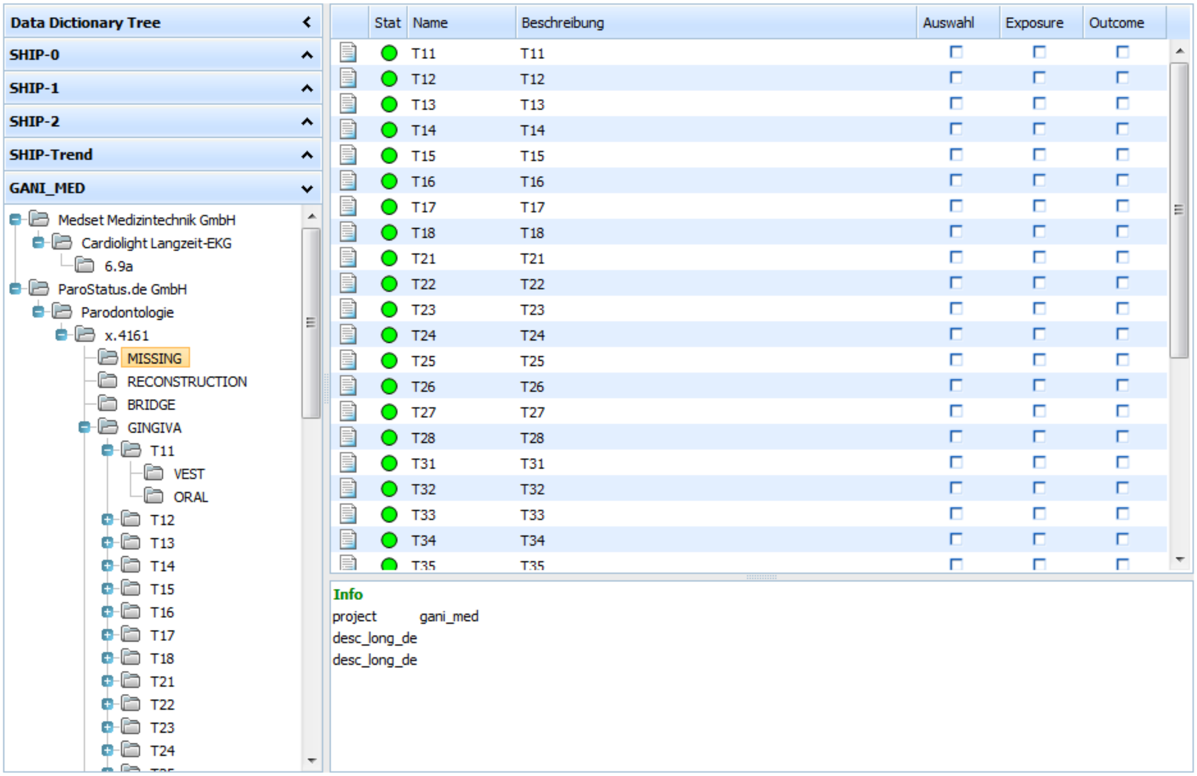
Figure 6: Exemplary *high level* structure of the values, observations and entities of the research database [21].

In the example shown above, there is a case number *FallNr: 123* and an order id *AuftragsNr.: 456* that connect e.g. a value called *hdlch* (HDL cholesterol) and two entities. The entities are the laboratory information system *Swisslab-IKCH* and a pseudonymized patient *X1AZ9T3QQE*. The actual observation values of *hdlch* and the gender and birthyear of the patient are stored by stacking them together.

The above mentioned abstract model is itself stored in an EAV model in the research database. It is important to note that this generic model contains an entity, yet this entity is not the EAV entity. The generic model is put on top of an underlying EAV database schema.

3.1.3 Development of the web application at the transfer unit

The current developmental version of the web application at the transfer unit looks as follows [22]:



Data Dictionary Tree

SHIP-0

SHIP-1

SHIP-2

SHIP-Trend

GANI_MED

- Medset Medizintechnik GmbH
  - Cardiolight Langzeit-EKG
    - 6.9a
- ParoStatus.de GmbH
  - Parodontologie
    - x.4161
      - MISSING
      - RECONSTRUCTION
      - BRIDGE
      - GINGIVA
        - T11
          - VEST
          - ORAL
        - T12
        - T13
        - T14
        - T15
        - T16
        - T17
        - T18
        - T21
        - T22
        - T23
        - T24

Stat	Name	Beschreibung	Auswahl	Exposure	Outcome
●	T11	T11	□	□	□
●	T12	T12	□	□	□
●	T13	T13	□	□	□
●	T14	T14	□	□	□
●	T15	T15	□	□	□
●	T16	T16	□	□	□
●	T17	T17	□	□	□
●	T18	T18	□	□	□
●	T21	T21	□	□	□
●	T22	T22	□	□	□
●	T23	T23	□	□	□
●	T24	T24	□	□	□
●	T25	T25	□	□	□
●	T26	T26	□	□	□
●	T27	T27	□	□	□
●	T28	T28	□	□	□
●	T31	T31	□	□	□
●	T32	T32	□	□	□
●	T33	T33	□	□	□
●	T34	T34	□	□	□
●	T35	T35	□	□	□

Info

project gani_med

desc_long_de

desc_long_de

Figure 7: Current development version of the transfer unit's web application, showing the SHiP study data application view and the GANI_MED data dictionary items [22].

In the future, this web application will allow researchers to select whole branches of the metadata tree shown on the left. Later in the process, researchers can then apply for that selected data to get a medical data output generated by the transfer unit. It is possible to specify that the selection is to be considered an *outcome*, *exposition* or just as a *chosen* variable.

As a means of simplicity and practicability, researchers are expected not to select each and every variable individually. Instead, they would select a whole group of variables, e.g. the complete somatometric data, complete smoking status, all laboratory variables of a specific group, and others.

Yet, there is also the possibility to restrict this process by not allowing the user to simply select the whole metadata tree. It is instead considered to select branches on a lower, more specific level. That means for example, that researchers cannot select all existing ultrasound examinations in total. Instead, they would need to select a certain, more specific type of ultrasound examination group.

The researcher is also not expected to read each of the variable's possible values during the selection process. Rather, he would select all the variables of his interest and wait for the results later. The results could be of a tabulatory nature, or be in a format of a specific statistical software. The actual possible values will be then read later in the result data.

3.1.4 BPMN process of the data application at the transfer unit

Below is a simplified *Business Process Modelling Notation* (BPMN) diagram of the application process that is running when a researcher submits an application (Figure 8, [23]). A detailed BPMN model of this process can be found in the appendix of this thesis (Figure 25, [23]).

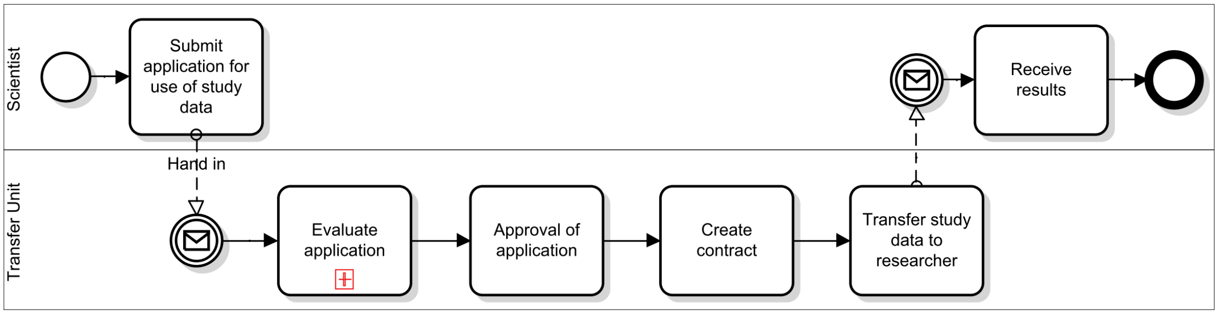


Figure 8: Simplified BPMN data application process at the transfer unit ([23], edited for simplification).

A researcher issues an application for the use of study data. That query will be checked by a review board at the transfer unit. After the query has been approved, a contract will be created for the use of that data. Then the study data is transferred to the researcher.

A more detailed view of the internal process of the data application can also be found in the appendix. There, the internal process of evaluating the researcher's request at the transfer unit is demonstrated (Figure 24, [24]).

3.1.5 The generic metadata repository G2D2

3.1.5.1 General explanations

G2D2 is a metadata object model developed at the ICM at Greifswald. It is used for working with the data dictionary of GANI_MED and allows to import and export metadata. Different source and target formats are supported. G2D2 contains *Data Transfer Object*s (DTOs), which could be loaded with metadata or where metadata can be exported from.

The advantage of this approach is the possibility of adding new import and export methods in the future. That makes it independent from the technology or formats used. Due to this general purpose usability, it can be called a generic model.

Below is a simplified model of the G2D2 model (Figure 9, [25]). Detailed models of G2D2 can be found in the appendix (Figures 26 and 27, [25]).

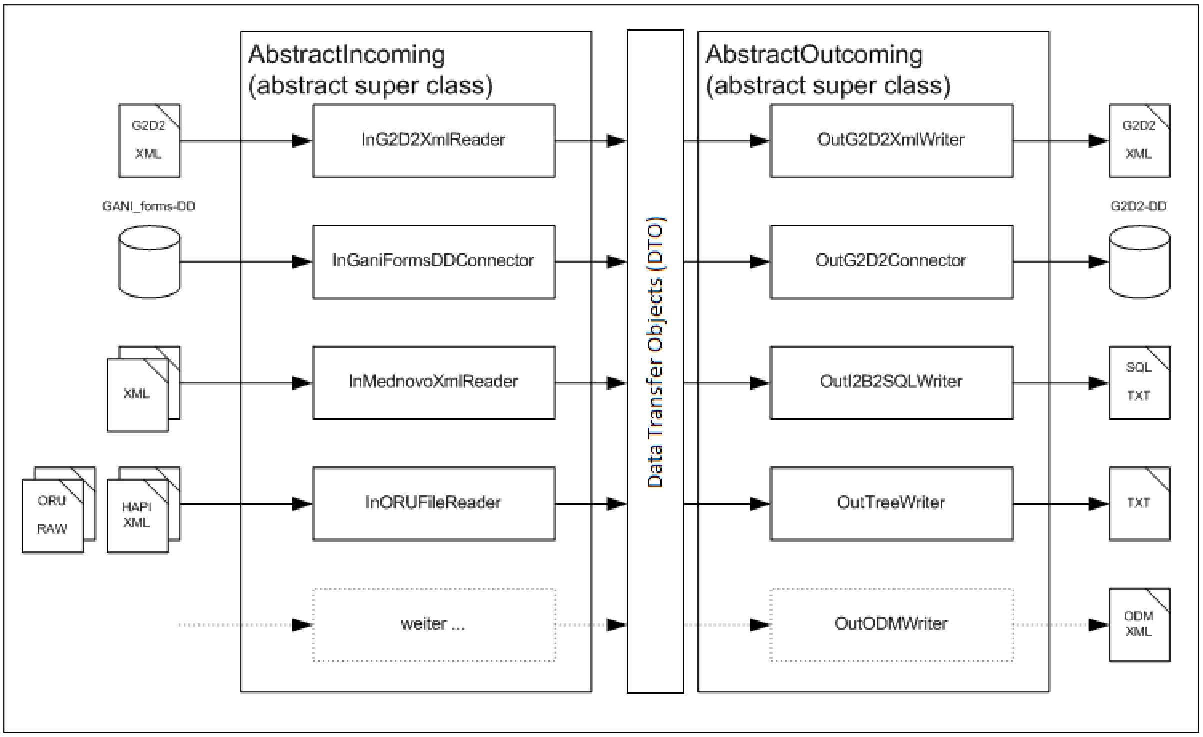


Figure 9: G2D2 DTO model ([25], edited for simplification).

3.2 Applied tools

This section contains the applied methods that were used to solve the problem. These methods were applied considering the before mentioned environmental conditions.

3.2.1 Software technology

In the following sections, the software technology and techniques that played a role in this thesis are shortly explained.

3.2.1.1 Eclipse

Eclipse is an *Integrated Development Environment* (IDE) e.g. for the Java programming language. Eclipse is licensed under the *Eclipse Public License* and is free open source software [26].

Eclipse is the IDE used in this thesis to develop the i2b2 import routine.

3.2.1.2 Aptana Studio 3

Aptana Studio 3 is an Eclipse *Rich Client Platform* (RCP) application that allows e.g. for the development of webapplications using JavaScript and is used for the development of the i2b2 webclient customization. Aptana Studio 3 is open source software and provided solely under the *GNU's Not Unix* (GNU) General Public License [27].

3.2.1.3 Java

Java is an object oriented programming language. Java allows to write programming code which runs in a virtual machine. This means, that it can be run on different hardware and software architectures without the need of recompilation of the underlying programming code [28][29].

Java is the programming language that is used to develop the i2b2 import routine in this thesis.

3.2.1.4 JavaScript

The i2b2 webclient is using JavaScript to provide its functionality, which is a script language that is interpreted on the client's computer. JavaScript is primarily used in webbrowsers, but becomes significant also in serverside programming or clientside desktop software development [30].

JavaScript is the programming language that is used by the i2b2 webclient.

3.2.1.5 FileSaver.js

FileSaver.js is a JavaScript based tool to generate files on a local webclient. These files can then be downloaded by a user. This is useful if there needs to be a file generated on the client without the use of a server. FileSaver.js is published under the *MIT/X11* license.

FileSaver.js is used in this thesis to generate a local text file on the client of a researcher (Section 4.2.6).

3.2.1.6 Oracle XE

In order to store the i2b2 data, an Oracle XE database server is used in the i2b2 instance. Oracle XE is the free-to-use edition of Oracle's RDBMS. This edition is limited in its abilities, it provides a limited amount of RAM to be used, harddisk storage and CPU cores that may be addressed when running the server [31].

i2b2 can also use external Oracle database systems, which may have better performance.

3.2.1.7 Ubuntu operating system

Ubuntu is an operating system derived from the Debian Linux distribution and is one of the most widely used Linux operating systems for both desktop and server systems [32].

The Ubuntu operating system is used to run the i2b2 application and the Oracle XE database system.

3.2.1.8 UML

The *Unified Modeling Language* (UML) allows to describe software systems in a variety of scenarios, such as sequence diagrams, activity diagrams and others. Through UML, a software architect documents and explains the functionality and architecture of e.g. a program. This is considered more convenient than reading the textual documentation or the code, as it uses graphical abstraction as a means of overview [33].

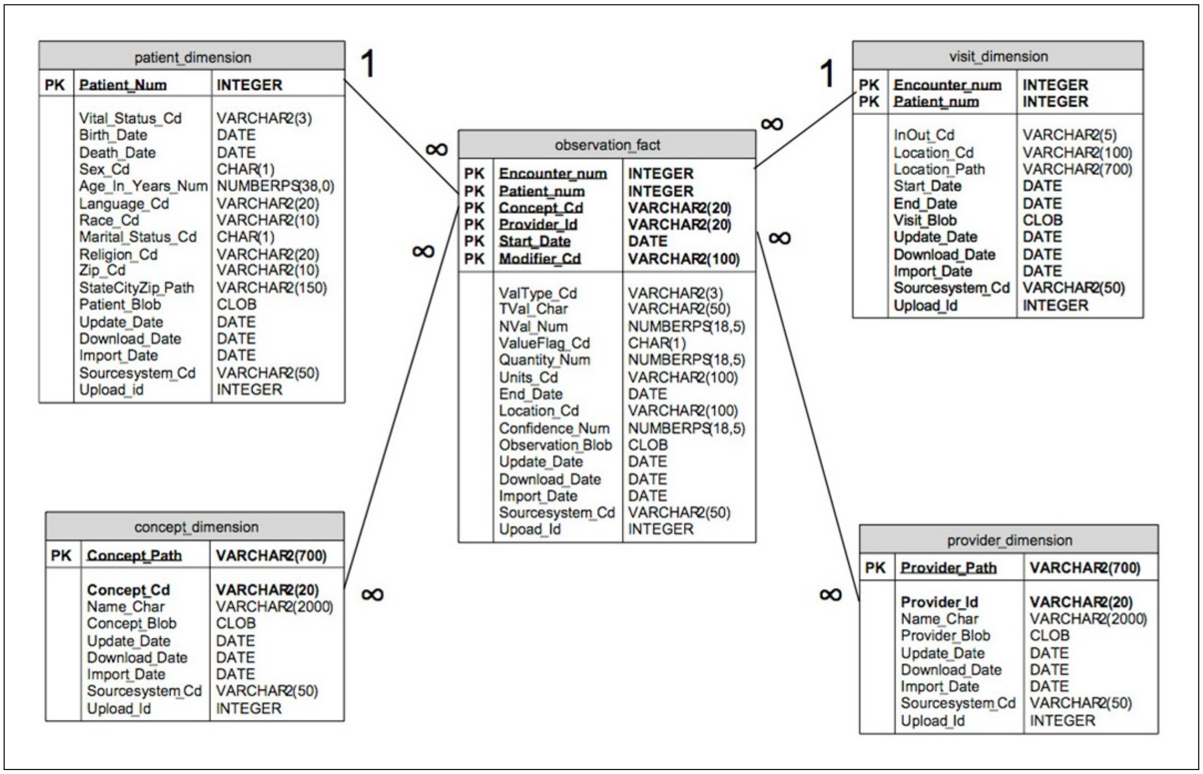
UML is used in this thesis to explain the software architecture and to visualize different models in the appendix.

3.2.1.9 i2b2

i2b2 is a *National Institutes of Health* (NIH) funded framework for storing and querying medical data through either a webclient or a local RCP application, called the i2b2 Workbench [34]. It is made of parts that are called i2b2 cells, which encapsulate a variety of functionality. These cells communicate with other cells through webservices, e.g. *Simple Object Access Protocol* (SOAP) messages. Through this encapsulated approach, one can develop new functionality that is integrated with the other cells. As a whole, these cells make up the so called i2b2 hive [35].

i2b2 is used as the secondary data warehouse in this thesis.

The i2b2 software is Java based and offers the possibility of importing data into a database that consists of parts for an ontology, for observational data, as well as for patients using a classical data warehouse star schema [36]:



patient_dimension

PK	**Patient_Num**	**INTEGER**
	Vital_Status_Cd	VARCHAR2(3)
	Birth_Date	DATE
	Death_Date	DATE
	Sex_Cd	CHAR(1)
	Age_In_Years_Num	NUMBERPS(38,0)
	Language_Cd	VARCHAR2(20)
	Race_Cd	VARCHAR2(10)
	Marital_Status_Cd	CHAR(1)
	Religion_Cd	VARCHAR2(20)
	Zip_Cd	VARCHAR2(10)
	StateCityZip_Path	VARCHAR2(150)
	Patient_Blob	CLOB
	Update_Date	DATE
	Download_Date	DATE
	Import_Date	DATE
	Sourcesystem_Cd	VARCHAR2(50)
	Upload_id	INTEGER

observation_fact

PK	**Encounter_num**	**INTEGER**
PK	**Patient_num**	**INTEGER**
PK	**Concept_Cd**	**VARCHAR2(20)**
PK	**Provider_Id**	**VARCHAR2(20)**
PK	**Start_Date**	**DATE**
PK	**Modifier_Cd**	**VARCHAR2(100)**
	ValType_Cd	VARCHAR2(3)
	TVal_Char	VARCHAR2(50)
	NVal_Num	NUMBERPS(18,5)
	ValueFlag_Cd	CHAR(1)
	Quantity_Num	NUMBERPS(18,5)
	Units_Cd	VARCHAR2(100)
	End_Date	DATE
	Location_Cd	VARCHAR2(100)
	Confidence_Num	NUMBERPS(18,5)
	Observation_Blob	CLOB
	Update_Date	DATE
	Download_Date	DATE
	Import_Date	DATE
	Sourcesystem_Cd	VARCHAR2(50)
	Upoad_Id	INTEGER

visit_dimension

PK	**Encounter_num**	**INTEGER**
PK	**Patient_num**	**INTEGER**
	InOut_Cd	VARCHAR2(5)
	Location_Cd	VARCHAR2(100)
	Location_Path	VARCHAR2(700)
	Start_Date	DATE
	End_Date	DATE
	Visit_Blob	CLOB
	Update_Date	DATE
	Download_Date	DATE
	Import_Date	DATE
	Sourcesystem_Cd	VARCHAR2(50)
	Upload_Id	INTEGER

concept_dimension

PK	**Concept_Path**	**VARCHAR2(700)**
	Concept_Cd	**VARCHAR2(20)**
	Name_Char	VARCHAR2(2000)
	Concept_Blob	CLOB
	Update_Date	DATE
	Download_Date	DATE
	Import_Date	DATE
	Sourcesystem_Cd	VARCHAR2(50)
	Upload_Id	INTEGER

provider_dimension

PK	**Provider_Path**	**VARCHAR2(700)**
	Provider_Id	**VARCHAR2(20)**
	Name_Char	VARCHAR2(2000)
	Provider_Blob	CLOB
	Update_Date	DATE
	Download_Date	DATE
	Import_Date	DATE
	Sourcesystem_Cd	VARCHAR2(50)
	Upload_Id	INTEGER

Figure 10: The i2b2 hive data model star schema, showing the core tables [36].

Besides the server side application and the database, i2b2 offers the possibility to query the database by using either the i2b2 Workbench or the i2b2 webclient. Due to the fact that the webclient doesn't need any installation on the user side, the webclient can be considered more convenient for the user, especially when there are many different researchers residing in different institutions [37]. A screenshot of the original i2b2 webclient can be found in the appendix (Figure 22).

3.2.1.10 i2b2 Wizard

The i2b2 Wizard in version 1.4.3 is a stable result of an ongoing research project conducted at Erlangen University, which greatly simplifies the process of installing and configuring an i2b2 instance. The i2b2 Wizard is part of the *Integrated Data Repository Toolkit* (IDRT) [38]. The wizard is used in the version that has been introduced at the first *European i2b2 Academic User Group (AUG) Workshop in Erlangen* in March, 2013. Among other i2b2 related applications, it utilizes an Oracle XE 10.2 database server and is made as a sophisticated Linux shell script using the Linux tool *dialog* [39].

3.3 Approach

This section describes the approach of solving the problem considering and using the before mentioned environment and applied tools.

3.3.1 Metadata import

The actual i2b2 metadata import tool[1] is programmed in Java and is making use of the G2D2 object model.

The G2D2 metadata is used by the metadata importing routine which reads the G2D2 files into a Java based DTO model. This model can then be accessed to extract metadata for the i2b2 import. G2D2 serves as a many-to-many relationship between different source and target systems. The model could be loaded by many different sources, as well as databases or a file based system. It may then be used to write the data to a target database, or other targets. In other words, this object model serves as a converter between a metadata source and target format.

For further improvement of the G2D2 metadata contents, there is the possibility of adding certain metadata attributes. These do not change the structure of G2D2 but enhance it by metadata needed e.g. by i2b2. An example may be an attribute that serves as an i2b2-specific display name for that metadata item.

[1] Developed mainly at Heidelberg University Hospital. An initial version of the metadata import tool including the DTO model and a recursive method to read the DTO model was delivered by the ICM.

The list of currently implemented, additional G2D2 attributes is shown in the below table (Table 2):

Attribute	Functionality
i2b2.ontology.visible	Set an i2b2 item visible or invisible in the ontology
i2b2.ontology.name	Define the title of the item shown to the user
i2b2.ontology.type	Datatype of the i2b2 item
i2b2.ontology.orderid	Integer that allows to sort items in the i2b2 ontology (prefix to item name)

Table 2: Additional G2D2 attributes for i2b2 ontology items.

The metadata importing tool is then making use of an optional import routine for observational data. That routine imports actual observational data from the research platform database into i2b2. The data importing routine is developed as a part of this thesis.

3.3.1.1 Software architecture

By running the import program, the software creates the necessary Java objects called InG2D2XmlReader[2] for reading the G2D2 *Extensible Markup Language* (XML) into the DTO model (Figures 9 and 26, 27), as well as the *OutI2B2SQLConnector*[3]. The *OutI2B2SQLConnector* then matches the metadata and potential observational data and writes them into the i2b2 database.

The *OutI2B2SQLConnector* makes use of a local configuration text file. That file contains the complete connection information about the research platform and the i2b2 target database. Both databases are considered to be Oracle instances of at least version 10.2.

The contents of the configuration text file are shown below (Listing 1):

```
# i2b2 host
oracle.db.ip = xxx.xxx.xxx.xxx
oracle.db.port = 1521
oracle.db.name = i2b2_sid
oracle.db.user = i2b2_user
```

[2] The InG2D2XmlReader was fully programmed at the ICM at Greifswald.

[3] The OutI2B2SQLConnector was developed for the most part at Heidelberg University Hospital.

```
oracle.db.pw = i2b2_password
oracle.db.db_prefix = i2b2_schema

# Research database
FoPlaDB.db.ip = xxx.xxx.xxx.xxx
FoPlaDB.db.port = 1521
FoPlaDB.db.name = xyzDB
FoPlaDB.db.user = xyzUser
FoPlaDB.db.pw = the_password

# Data cleansing logfiles
DataCleansingError.log = ./DataCleansingError.log
DataCleansingCorrect.log = ./DataCleansingCorrect.log
```

Listing 1: Configuration file for the *OutI2B2SQLConnector*

The *OutI2B2SQLConnector* is, as the name suggests, specifically designed to work solely with i2b2 as a target and is not meant to deal with any other database.

A high level view of the import process is shown below (Figure 11):

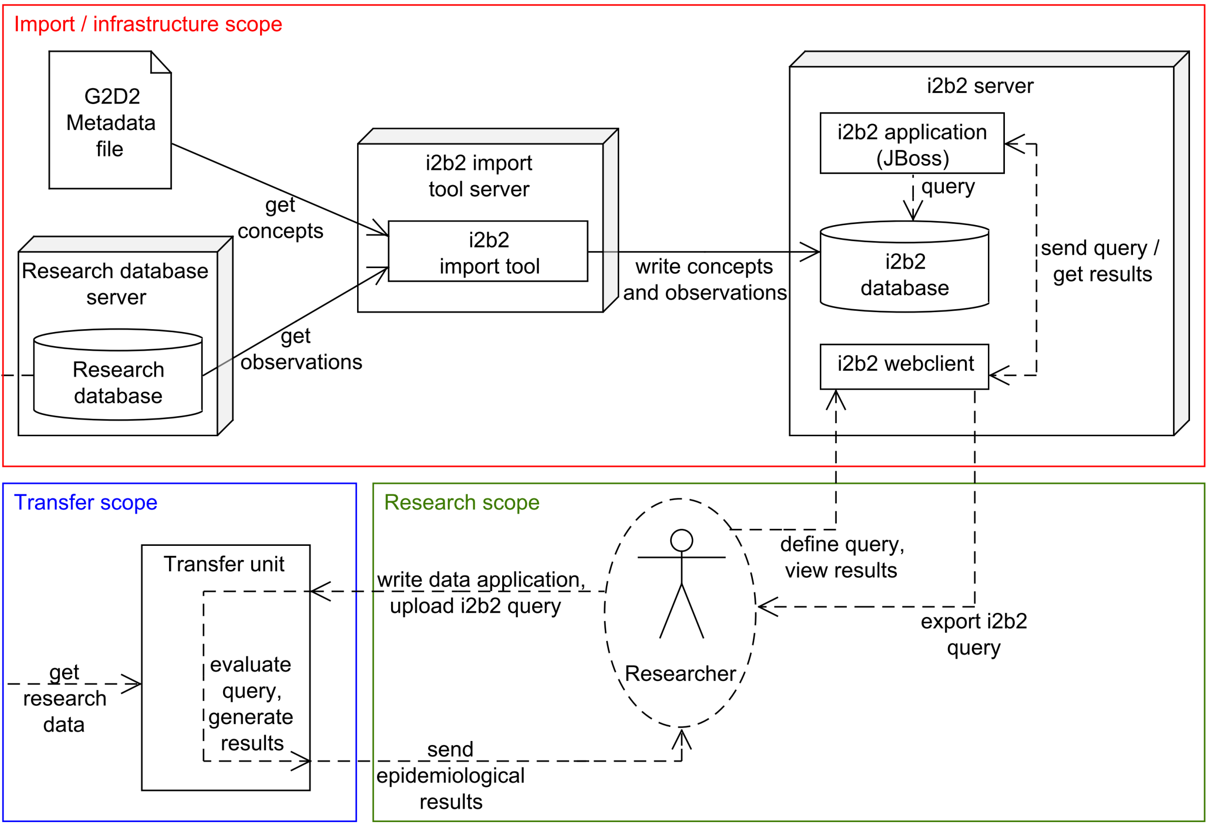


Figure 11: Overview of the i2b2 import routine architecture.

3.3.2 Planned i2b2 webclient customization

In order to let a researcher define a query for the transfer unit, the i2b2 webclient is customized. The customized webclient has a new tab next to the *Query Tool* tab. There, a researcher will be able to define a query, which will be transmitted to the transfer unit. The new tab allows to define metadata items as being an *Exposure*, an *Outcome*, or an *Other* variable. The *Other* variables can contain confounding variables, as well as simply chosen variables that are neither an outcome, nor an exposure.

4 Results

In this section, the results of this thesis related to the before mentioned methods section are described. The results related to the goals of the thesis are demonstrated.

4.1 First goal: Integration of metadata and research data into i2b2

The data integration of the G2D2 metadata and the research data is split into two parts. Those are the import of the metadata into the i2b2 ontology and the import of the research data into the i2b2 star schema.

The i2b2 ontology is one table in the i2b2 database and follows the below SQL table schema (Table 3, [40]):

COLUMN NAME	DATA TYPE (ORACLE)
C_HLEVEL	INT
C_FULLNAME	VARCHAR2(700)
C_NAME	VARCHAR2(2000)
C_SYNONYM_CD	CHAR(1)
C_VISUALATTRIBUTES	CHAR(3)
C_TOTALNUM	INT
C_BASECODE	VARCHAR2(50)
C_METADATAXML	CLOB
C_FACTTABLECOLUMN	VARCHAR2(50)
C_TABLENAME	VARCHAR2(50)
C_COLUMNNAME	VARCHAR2(50)
C_COLUMNDATATYPE	VARCHAR2(50)
C_OPERATOR	VARCHAR2(10)
C_DIMCODE	VARCHAR2(700)
C_COMMENT	CLOB
C_TOOLTIP	VARCHAR2(900)
UPDATE_DATE	DATE
DOWNLOAD_DATE	DATE
IMPORT_DATE	DATE
SOURCESYSTEM_CD	VARCHAR2(50)
VALUETYPE_CD	VARCHAR2(50)

Table 3: i2b2 ontology table [40]

The ontology table may itself be named freely, yet it has to be registered in the TABLE_ACCESS table in i2b2. This allows to use it with an i2b2 client. The ontology table is the table the metadata tree is rendered from on an i2b2 client. Every item in the ontology table contains the entire hierarchical structure of the metadata item in the C_FULLNAME field.

The G2D2 metadata is processed by the data import tool and inserted into the i2b2 ontology table. The link between the ontology table and the OBSERVATION_FACT table, which contains the actual observations, is done over the CONCEPT_DIMENSION table, that looks as follows (Table 4 [40]:

COLUMN NAME	DATA TYPE (ORACLE)
CONCEPT_PATH	VARCHAR(700)
CONCEPT_CD	VARCHAR(50)
NAME_CHAR	VARCHAR(2000)
CONCEPT_BLOB	TEXT
UPDATE_DATE	DATETIME
DOWNLOAD_DATE	DATETIME
IMPORT_DATE	DATETIME
SOURCESYSTEM_CD	VARCHAR(50)
UPLOAD_ID	INT

Table 4: i2b2 CONCEPT_DIMENSION table [40]

It is important to note that the C_BASECODE in the ontology table actually is the CONCEPT_CD in the CONCEPT_DIMENSION. That field is used to link the ontology with the i2b2 star schema. The CONCEPT_CD field is also found in the OBSERVATION_FACT table in order to run the internal SQL query in i2b2, if the user uses the Query Tool of e.g. the webclient. The CONCEPT_CD is therefore the link between the ontology table, the CONCEPT_DIMENSION and the OBSERVATION_FACT table.

Below is an example observation from the research database (Table 5).

Item	Value
DDREF_CACHE.PUBLIC_DDREF_ID	swisslab.ikch-n_src_vers-HGB,E
DDREF_CACHE.URL	\GANI_MED\...\swisslab\ikch\HGB,E\
ENTITY.IDENTIFIER	ABC1234
OBSERVATION.CASE_ID	4321CBA
OBSERVATION.OBSERVATION_ID	ZYX987
RAW_DATA.IDENTIFIER	SWISSLAB_123.XML
VALUE_HISTORY.VALUE	9
VALUE_ELEMENT_VALUE.UNIT	mmol/l
VALUE_HISTORY.TIMESTAMP	2012-10-25 09:24:00.0

Table 5: Example observation from the research database (single row)

The before shown exemplary observation from the research database is imported into i2b2. The mapping between the research database and i2b2 is explained below (Figure 12).

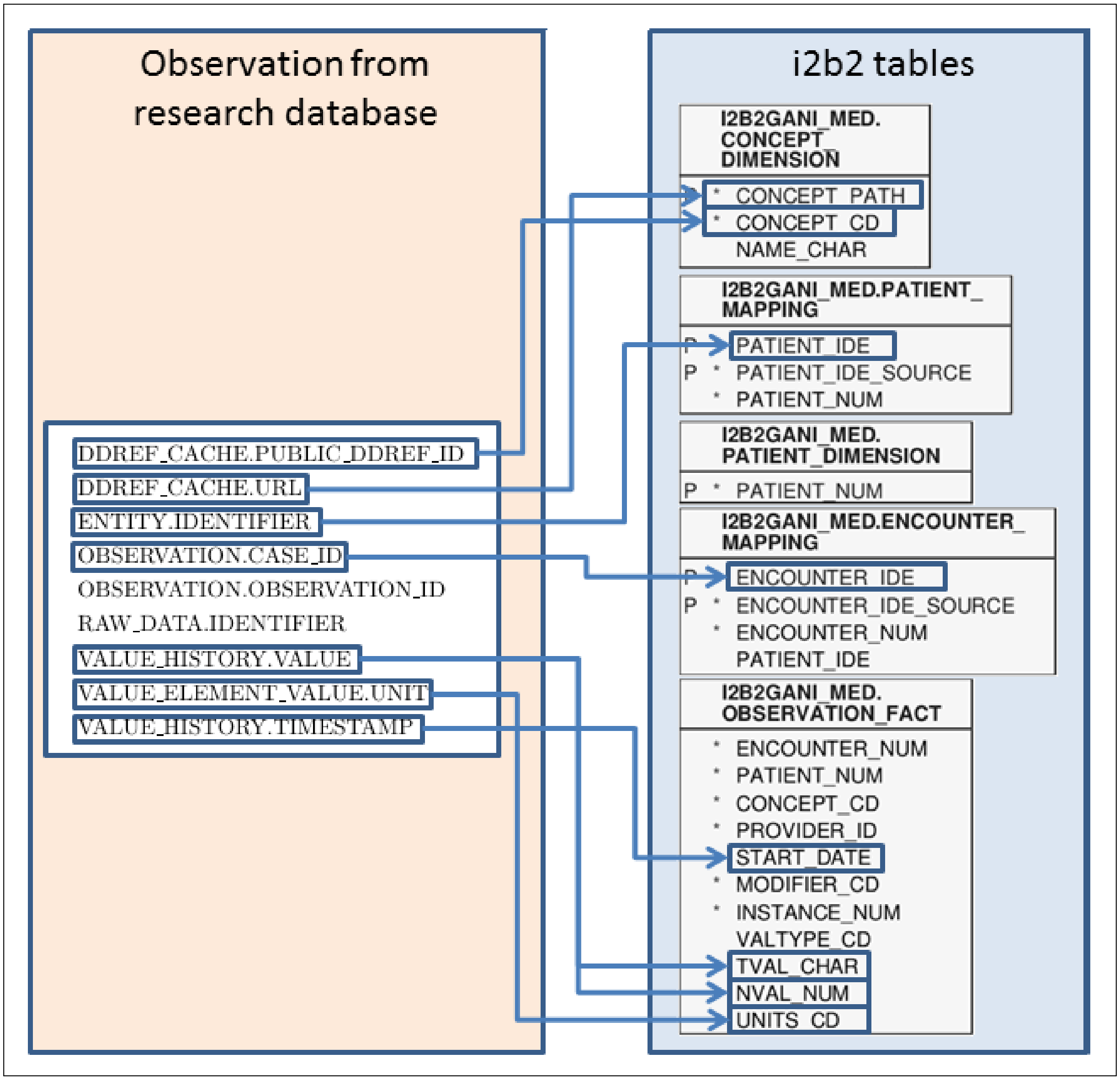


Figure 12: Mapping of the research database query with the i2b2 database

Below is an example G2D2 metadata item (hemoglobin). (Table 6). The metadata contains references to attributes as well as to parents or lingual definitions.

XML attribute	Value
Name	HBG,E
PublicDDReferenceID	swisslab.ikch-n_src_vers-HGB,E
OID	E.167
XML Element	Value(s)
ElementTypeRef	ET.6: ”oru_item”
ParentElementRef	E.110: ”source.root”
AttributeRef	A.74: ”mmol/l”
AttributeRef	A.117: ”7.4 - 10.0—8.6 - 11.2—7.4 - 11.2”
AttributeRef	A.1156: ”1”
AttributeRef	A.1157: ”3”
AttributeRef	A.2532: ”HGB Hämoglobin”
DescriptionLanguageResourceRef	LR.177: HGB Hämoglobin Material: 0.2 ml EDTA-Blut Referenzbereich: bis 7 Tage 9.3 - 14.9 mmol/l 8 bis 21 Tage 7.9 - 11.6 mmol/l 22 bis 60 Tage 5.6 - 10.3 mmol/l 61 bis 210 Tage 6.0 - 8.0 mmol/l bis 3 Jahre 6.5 - 8.1 mmol/l 4 bis 5 Jahre 6.9 - 8.9 mmol/l 6 bis 11 Jahre 7.4 - 9.1 mmol/l Frauen 12 Jahre bis ins Alter 7.4 - 10.0 mmol/l Männer 12 Jahre bis ins Alter 8.6 - 11.2 mmol/l allgemein 7.4 - 11.2 mmol/l

Table 6: G2D2 element definition for *ElementDef*

The before shown metadata item (Table 6) is mapped to i2b2 by using the *OutI2B2SQLConnector*.

The mapping of the G2D2 metadata to the i2b2 ontology is shown below (Figure 13):

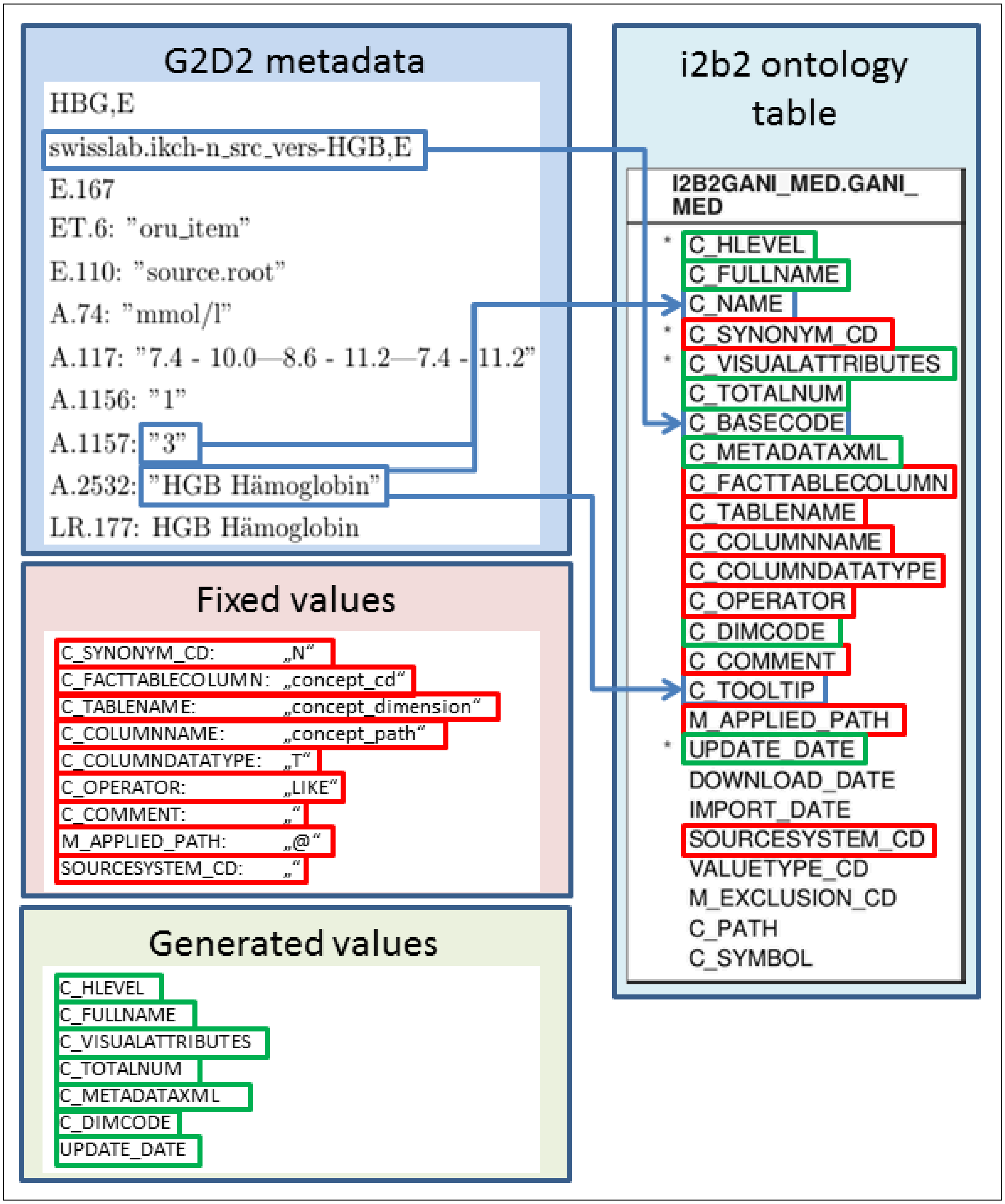


Figure 13: Mapping of G2D2 to the i2b2 ontology table

Only the columns C_NAME, C_BASECODE and the C_TOOLTIP contain unchanged data from the source metadata file (Figure 13). The other columns are filled with data that the *OutI2B2SQLConnector* generates internally using the G2D2 metadata (marked green). These values are generated as follows:

- C_HLEVEL: Hierarchical level of the G2D2 metadata item (Integer$\geq$0).
- C_FULLNAME: Full hierarchical path of the item.
- C_VISUALATTRIBUTES: Depending on the level of the item, the item is either a *container, folder* or a *leaf* item. *Containers* can't be queried for, the other two can. *Leafs* may allow to specify a unit and a numerical range of the item.
- C_TOTALNUM: Total number of the observations that this item has. May increase query performance.
- C_METADATAXML: XML metadata information necessary to allow to specify a numerical constraint and to select a unit for the item.
- C_DIMCODE: Full hierarchical path of the item.
- UPDATE_DATE: The date the data was updated (imported).

Other values are hard coded in the *OutI2B2SQLConnector* (marked red). Later versions of the software may get these values also from the metadata. The unmarked values in the ontology table are *null* values.

The below diagram describes the import process of the G2D2 metadata to the i2b2 ontology table (Figure 14).

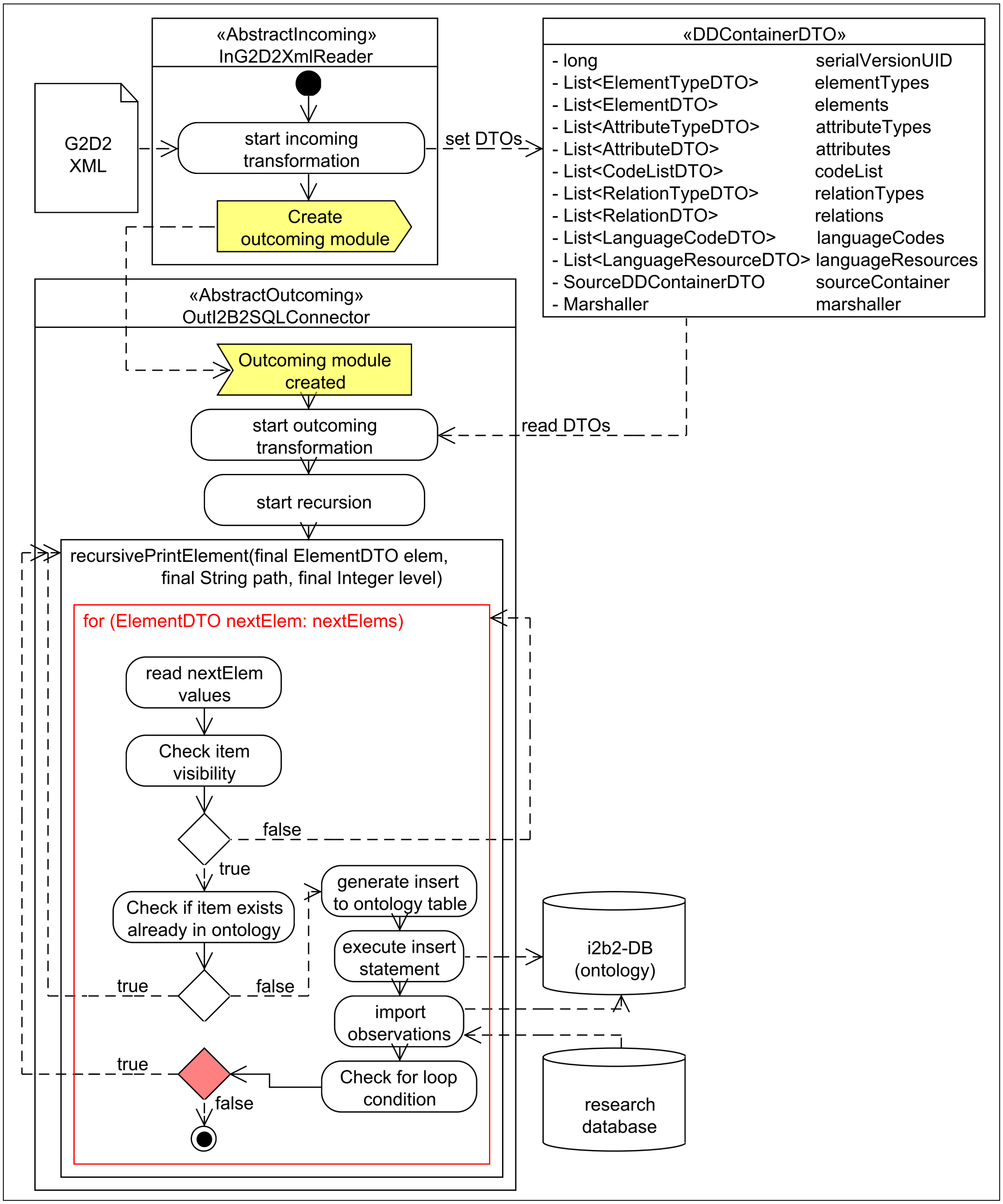

G2D2
XML
«AbstractIncoming»
InG2D2XmlReader
start incoming transformation
Create outcoming module
set DTOs
«DDContainerDTO»
- long serialVersionUID
- List<ElementTypeDTO> elementTypes
- List<ElementDTO> elements
- List<AttributeTypeDTO> attributeTypes
- List<AttributeDTO> attributes
- List<CodeListDTO> codeList
- List<RelationTypeDTO> relationTypes
- List<RelationDTO> relations
- List<LanguageCodeDTO> languageCodes
- List<LanguageResourceDTO> languageResources
- SourceDDContainerDTO sourceContainer
- Marshaller marshaller
«AbstractOutcoming»
OutI2B2SQLConnector
Outcoming module created
start outcoming transformation
read DTOs
start recursion
recursivePrintElement(final ElementDTO elem, final String path, final Integer level)
for (ElementDTO nextElem: nextElems)
read nextElem values
Check item visibility
false
true
Check if item exists already in ontology
true
false
generate insert to ontology table
execute insert statement
import observations
Check for loop condition
true
false
i2b2-DB (ontology)
research database

Figure 14: Importing the G2D2 metadata into the i2b2 ontology table

The process shown before in Figure 14 imports a G2D2 file and sets the necessary DTO values. After that, the *OutI2B2SQLConnector* is created, reading the DTO data and starting a recursive method to iterate over that data.

The recursion contains a loop that iterates over the *ElementDTO*s of the current hierarchical level and checks, whether this element is visible. If it is and it is not yet in the ontology, it will be inserted. Optionally, the observational data is imported, too.

If it is not a visible item, the loop will pick the next *ElementDTO* and start over. If the item exists in the ontology, or the import process for the item is over, the recursive method is called again. The for loop is exited there and the process tries to find visible and non existent items in the next level. The path variable is used to generate the C_FULLNAME and C_DIMCODE in the i2b2 ontology table, as it is the result of a recursive iteration over the DTO hierarchy.

4.1.1 Importing research data into the i2b2 star schema

The method of importing research data into the i2b2 star schema is optionally called by the metadata import method. It allows to skip data imports for certain metadata items, if necessary.

The i2b2 data import needs to know,

- which ontology item it should insert into the star schema,
- which values in the research database are legal for the import (since the research database may contain unclean data),
- and whether the item should be treated as a numerical or a textual item.

The i2b2 data import method will then retrieve the research data that fits the given identifier (the public_ddref_id of that metadata item from G2D2) through a JDBC Oracle connection and will generate lists of SQL statements for inserting the necessary data into the i2b2 star schema.

The tables that are populated with data are:

- The ontology table called GANIMED

- CONCEPT_DIMENSION
- PATIENT_DIMENSION
- PATIENT_MAPPING
- ENCOUNTER_MAPPING
- OBSERVATION_FACT

To fill these tables is sufficient in order to allow the user to count the amount of patients fitting his i2b2 query. To populate the ENCOUNTER_DIMENSION is not necessary for the purpose of counting the patients. Yet, it may be useful for later purposes during the GANI_MED project and is therefore included.

The PATIENT_DIMENSION table looks as follows (Table 7):

COLUMN NAME	DATA TYPE (ORACLE)
PATIENT_NUM	INT
VITAL_STATUS_CD	VARCHAR(50)
BIRTH_DATE	DATETIME
DEATH_DATE	DATETIME
SEX_CD	VARCHAR(50)
AGE_IN_YEARS_NUM	INT
LANGUAGE_CD	VARCHAR(50)
RACE_CD	VARCHAR(50)
MARITAL_STATUS_CD	VARCHAR(50)
RELIGION_CD	VARCHAR(50)
ZIP_CD	VARCHAR(10)
STATECITYZIP_PATH	VARCHAR(700)
PATIENT_BLOB	TEXT
UPDATE_DATE	DATETIME
DOWNLOAD_DATE	DATETIME
IMPORT_DATE	DATETIME
SOURCESYSTEM_CD	VARCHAR(50)
UPLOAD_ID	INT

Table 7: i2b2 PATIENT_DIMENSION table [40]

In the current, developmental version of the data import routine, this table contains only an automatically incremented integer value for the field

PATIENT_NUM. This is sufficient for i2b2 to count the number of patients for a given query in the i2b2 webclient. In later versions, this table may be populated with additional information about the patient, to allow further analysis, like age breakdown, or others.

Below, the PATIENT_MAPPING table is introduced (Table 8):

COLUMN NAME	DATA TYPE (ORACLE)
PATIENT_IDE	VARCHAR(200)
PATIENT_IDE_SOURCE	VARCHAR(50)
PATIENT_NUM	INT
PATIENT_IDE_STATUS	VARCHAR(50)
UPDATE_DATE	DATETIME
DOWNLOAD_DATE	DATETIME
IMPORT_DATE	DATETIME
SOURCESYSTEM_CD	VARCHAR(50)
UPLOAD_ID	INT

Table 8: i2b2 PATIENT_MAPPING table [40]

This table contains a link to the PATIENT_DIMENSION by using the field *PATIENT_NUM* in this table. Here, the field *PATIENT_IDE* contains the patient identifier, or PID. The *E* in *...IDE* of that field stands for *encrypted*. Due to the fact, that the patient identifier is pseudonymized, the contents of this field can be considered *encrypted*.

The ENCOUNTER_MAPPING table looks as follows (Table 9):

COLUMN NAME	DATA TYPE (ORACLE)
ENCOUNTER_IDE	VARCHAR(200)
ENCOUNTER_IDE_SOURCE	VARCHAR(50)
ENCOUNTER_NUM	INT
PATIENT_IDE	VARCHAR(200)
PATIENT_IDE_SOURCE	VARCHAR(50)
ENCOUNTER_IDE_STATUS	VARCHAR(50)
UPLOAD_DATE	DATETIME
DOWNLOAD_DATE	DATETIME
IMPORT_DATE	DATETIME
SOURCESYSTEM_CD	VARCHAR(50)
UPLOAD_ID	INT

Table 9: i2b2 ENCOUNTER_MAPPING table [40]

This table allows to link a medical encounter to the patient. This table contains the *PATIENT_IDE* from the before mentioned tables PATIENT_DIMENSION and PATIENT_MAPPING. The field *ENCOUNTER _IDE* works like the field *PATIENT_IDE.* It contains a case number, that is encrypted, or as mentioned before, pseudonymized.

The OBSERVATION_FACT table is explained below (Table 10):

COLUMN NAME	DATA TYPE (ORACLE)
ENCOUNTER_NUM	INT
CONCEPT_CD	VARCHAR(50)
PROVIDER_ID	VARCHAR(50)
START_DATE	DATETIME
MODIFIER_CD	VARCHAR(50)
PATIENT_NUM	INT
VALTYPE_CD	VARCHAR(50)
TVAL_CHAR	VARCHAR(255)
NVAL_NUM	DECIMAL(18,5)
VALUEFLAG_CD	VARCHAR(50)
QUANTITY_NUM	DECIMAL(18,5)
UNITS_CD	VARCHAR(50)
END_DATE	DATETIME
LOCATION_CD	VARCHAR(50)
OBSERVATION_BLOB	TEXT
CONFIDENCE_NUM	DECIMAL(18,5)
UPDATE_DATE	DATETIME
DOWNLOAD_DATE	DATETIME
IMPORT_DATE	DATETIME
SOURCESYSTEM_CD	VARCHAR(50)
UPLOAD_ID	INT

Table 10: i2b2 OBSERVATION_FACT table [40]

This table contains the observations that are inserted into i2b2 by the import routine. Observations have a link to a patient (*PATIENT_NUM)* and an encounter (*ENCOUNTER_NUM*). The *START_DATE* contains the timestamp of the beginning of the observation. The *END_DATE* is not used here, but it indicates the time of the end of the observation. In order to link the observation to a concept, the *CONCEPT_CD* of the CONCEPT_DIMENSION table is contained here, too. From the CONCEPT_DIMENSION table, a link to the ontology table is possible, allowing to query the i2b2 database by a user using the i2b2 webclient.

The fields *TVAL_CHAR* and *NVAL_NUM* act as a storage for the actual value of the observation. If the value's datatype is textual, then the

TVAL_CHAR field is used to store that value. Otherwise, the *NVAL_NUM* field is used. The unit of the value can be stored in the *UNITS_CD* field.

The i2b2 data import method is shown in the below diagram. This optional method is called for every single metadata item for which a data import is required:

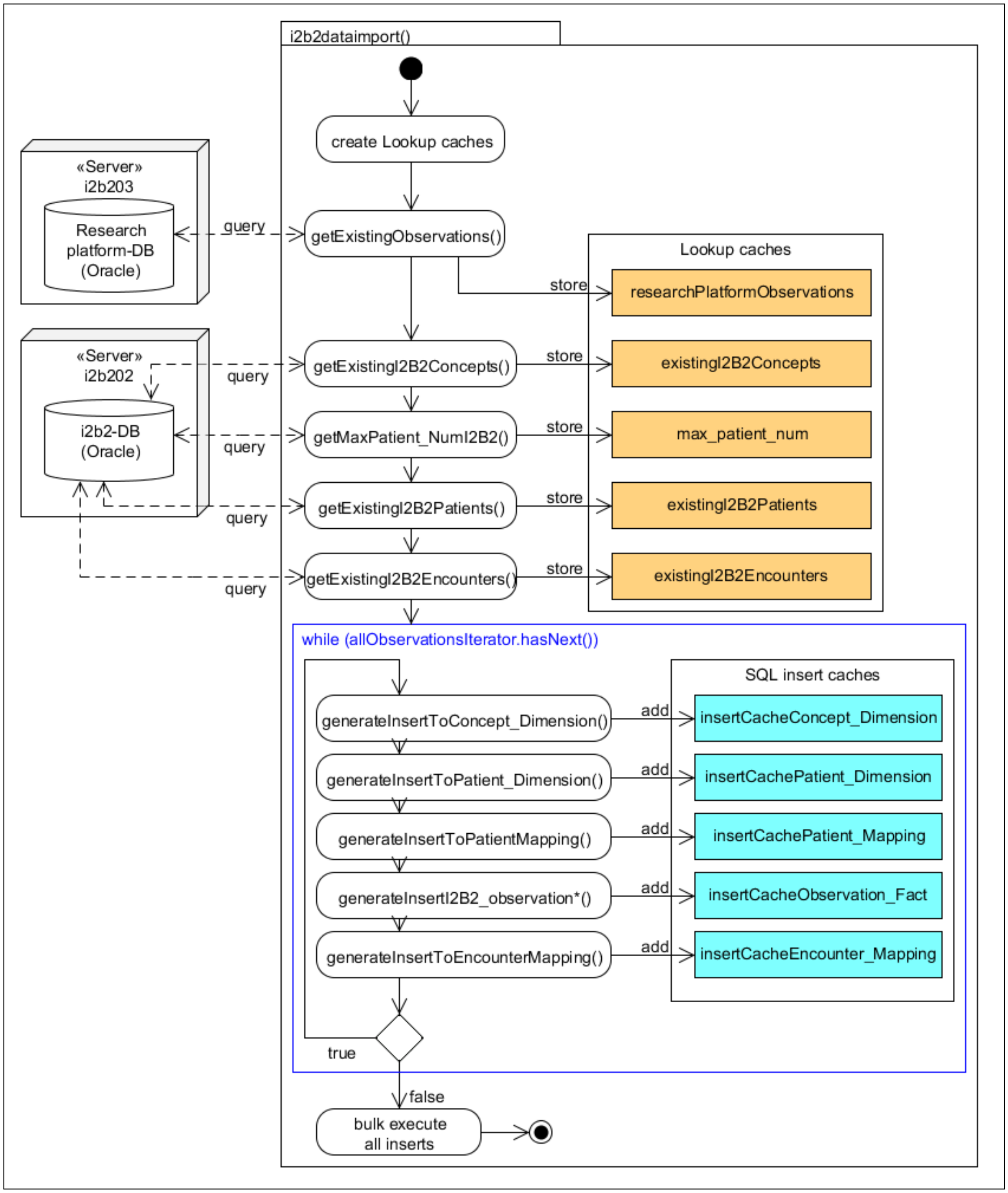

i2b2dataimport()
create Lookup caches
«Server»
i2b203
Research platform-DB (Oracle)
query
getExistingObservations()
Lookup caches
store
researchPlatformObservations
«Server»
i2b202
i2b2-DB (Oracle)
query
getExistingI2B2Concepts()
store
existingI2B2Concepts
query
getMaxPatient_NumI2B2()
store
max_patient_num
query
getExistingI2B2Patients()
store
existingI2B2Patients
query
getExistingI2B2Encounters()
store
existingI2B2Encounters
while (allObservationsIterator.hasNext())
SQL insert caches
generateInsertToConcept_Dimension()
add
insertCacheConcept_Dimension
generateInsertToPatient_Dimension()
add
insertCachePatient_Dimension
generateInsertToPatientMapping()
add
insertCachePatient_Mapping
generateInsertI2B2_observation*()
add
insertCacheObservation_Fact
generateInsertToEncounterMapping()
add
insertCacheEncounter_Mapping
true
false
bulk execute all inserts

Figure 15: i2b2 data import method

The idea behind this process is to use local caches of already existing data from i2b2 and from the research database. These caches should allow a minimal amount of database connections, which may increase the overall performance.

In the first part of the process, the caches are loaded by reading the existing observations for a given metadata item from the research database as well as the already existing relevant data from i2b2. Having that information, the below shown while loop can be used to generate insert statements from the local caches. These insert statements are intended to write the necessary i2b2 tables. The generated inserts are then executed in bulk at the end of the process.

The data import method has the following method header:

```
private int i2b2dataimport(String public_ddref_id, String concept_path,
    List<CodeListItemDTO> codeList)
```

i2b2dataimport

The public_ddref_id is the unique identifier for the given metadata item. The concept_path is the concept_path from the ontology table. The parameter *codeList* is a list of CodeListItemDTO objects from G2D2 which are used as a whitelist to identify legal values in the research database.

If a value is found in the research database which is not in the *codeList*, then that value is considered an illegal value and is not inserted into the i2b2 database. This whitelist check works only for textual values, since there is no check for illegal numerical values, e.g. values that are unreasonably high or low.

4.1.1.1 Data cleansing and quality assurance

Besides the data cleansing of pending or not useful data, the quality assurance personnel also cleans systematic errors in the data. These errors could arise through changed medical device configurations, or errors that were caused by the individual medical examiners.

Some systems also allow the users to add manually typed information to the database, if necessary. Due to the unstructured nature of that sort of comments, this data often can't be imported into the secondary data warehouse and needs to be cleaned. That means that it is either skipped from the import or mapped to the metadata.

This data cleansing will be part of the quality assurance at Greifswald. i2b2 will receive its data in a clean form from this quality assurance, in order to contain only that information in the database that is necessary for researchers. During the import of that data, there is another data cleansing step that relates to the i2b2 data schema. i2b2 distinguishes values that are textual or numerical. For that reason, a second independent cleansing process is implemented during the actual data import by using whitelists for each textual metadata item. Numerical values are not checked for unreasonable values, such as too high or too low values. This would mean that the import program needs to have a logic implemented for that purpose. Instead, the data that is imported into i2b2 should be clean beforehand.

4.1.2 Data cleansing logs

During the importing routine, a logging functionality is also implemented. There are two logfiles that explain, which items were inserted into i2b2, which values they have and how often they were inserted. One logfile works as the log of correctly inserted items, the other one contains the list of unclean data that was not inserted into i2b2.

One line in the log for correct values may look as follows:

```
91;swisslab.ikch-n_src_vers-HGB,E-1;6.2
```

This means that there were 91 inserts for hemoglobin that had the value 6.2 . The value's unit is not included.

The logfile for unclean data may contain this line:

```
124;swisslab.ikch-n_src_vers-HGB,E-1;k.Mat.
```

This entry means that there have been 124 inserts that were not executed because they contained a value that was not expected (non-numerical value for a numerical G2D2 item), in this case *k.Mat.* (”kein Material”, ger. for *no material*). This value should be excluded from the import, since a researcher is not interested in this kind of data. For a numerical metadata item, all values in the research database which are non-numerical are considered an erroneous value and will not be included in the i2b2 database after the import.

For a textual variable, there is a whitelist check for comparing the found value in the research database to G2D2 metadata whitelist items. This is done in order to decide, whether there should be an import of this dataset or not.

An example may be a patient's urine coloring that could be one of five different textual values, e.g. *Hellgelb, Gelb, Dunkelgelb, Braun, Rot* (ger. for *bright yellow, yellow, dark yellow, brown, red*). In this scenario, there may be a value found in the research database that is illegal, because it is not on the whitelist, e.g. *RED*.

One could argue that most likely the value *RED* should be imported as the german whitelist item *Rot* (engl. *red*). This would mean that there needs to be a logic implemented into the importing program which should be avoided due to the fact that *RED* may first of all be an abbreviation for something else, since it is written all in capital letters. Secondarily, this would mean that there is unnecessary logic introduced into the importing program. The intelligence of the whole process should rather remain in the metadata and research data, in other words, the input data needs to be cleaned before the import. Whether it would be legal for the database administrator to simply alter the values of *RED* into *Rot* inside the research database is also a question that needs to be considered.

The whitelists are contained in the G2D2 metadata and may be altered as necessary. After altering them, the whole import process has to start over to take effect.

The logfiles can help to find values that should be put on the whitelist or be removed from it. That means that the logfiles may act as a worklist for data management and data cleansing personnel that clean the research database from unwanted data, simply by working through the logfiles.

4.1.3 Import performance comparison

Below is a performance comparison between two different computers. Those computers executed the import routine for the laboratory information system *Swisslab*. In the first test, the machines imported only the metadata, whereas in the second test, they imported both, the metadata and the observational data. A *Virtual Machine* (VM) containing an i2b2 instance was started on each of the computers, as well as the import routine itself. In other words, the import program, i2b2 and the i2b2 database were on each of the physical machines. The research database was installed on the second computer. The reason for installing the research database on the second computer was the better performance of its hardware. Due to that, the time needed to query the research database was greatly reduced.

The hardware specification of the two devices is shown below (Table 11):

	Computer 1	**Computer 2**
Model	Lenovo ThinkPad T61	Dell Latitude E6430
Manufacturing year	2008	2013
Processor	Intel Core 2 Duo, 2 GHz	Intel Core i7 (4 cores + HT), 2,7 GHz
RAM	4 GB	16 GB
Hard drive	SSD: KINGSTON SV300S37A240G	SSD: LITEONIT LCS-256M6S
OS	Windows 7 Professional SP1	Windows 7 Professional SP1

Table 11: Hardware specification of computers used to measure the import performance

The performance of the above mentioned computers when executing the import program is shown below (Tables 12 and 13).

Machine: *Computer 1*	**Time to completion**	**Concepts inserted**	**Observations inserted**
Metadata import	28s	35	0
Metadata + observational data import	10min 11s	35	117658

Table 12: Result of the performance test for *Computer 1*, importing metadata and observational data, measuring the time from start to completion of the import.

Machine: *Computer 2*	**Time to completion**	**Concepts inserted**	**Observations inserted**
Metadata import	4s	35	0
Metadata + observational data import	3min 17s	35	117658

Table 13: Result of the performance test for *Computer 2*, importing metadata and observational data, measuring the time from start to completion of the import.

4.2 Second goal: Integration of the i2b2 webclient query

4.2.1 Requirements of query exchange of i2b2 with the transfer unit

The transfer unit's web application, where a researcher can issue an application for research data, is still in development. Yet, there are only few changes left to get the application to productive use. There needs to be an import of the G2D2 metadata to the web application. This work is about to be finished, only few minor changes are still necessary. This metadata import is necessary to work together e.g. with the i2b2 metadata export files.

In May 2013, there was an informal meeting with the man in charge of the transfer unit at Greifswald, where the developmental version of the web application was presented and possible user scenarios were discussed.

The functionality of the web application was explained by presenting the web application's GUI. It was discussed what intentions a researcher has when he or she visits the web application.

The general requirements derived from the given presentation and discussions for the query exchange format can be summarized as:

- Standardized exchange file format, machine readable
- Exchange file should be downloaded by the researcher from i2b2 and uploaded to the transfer unit
- Exchange file contains *Exposure, Outcome* and *Other* variables including the unique identifier and the hierarchical information of the item, as well as the inclusion and exclusion criteria from the i2b2 Query Tool

4.2.2 Planned i2b2 webclient customization

The following picture is a graphical mockup made from the i2b2 webclient screenshot. It shows the intended new tab and the three boxes for the *Exposure*, *Outcome* and *Other* variables. Below the three boxes are two new buttons to export and import the selected variables for exchanging the query with the research application form, or to restore a previous query.

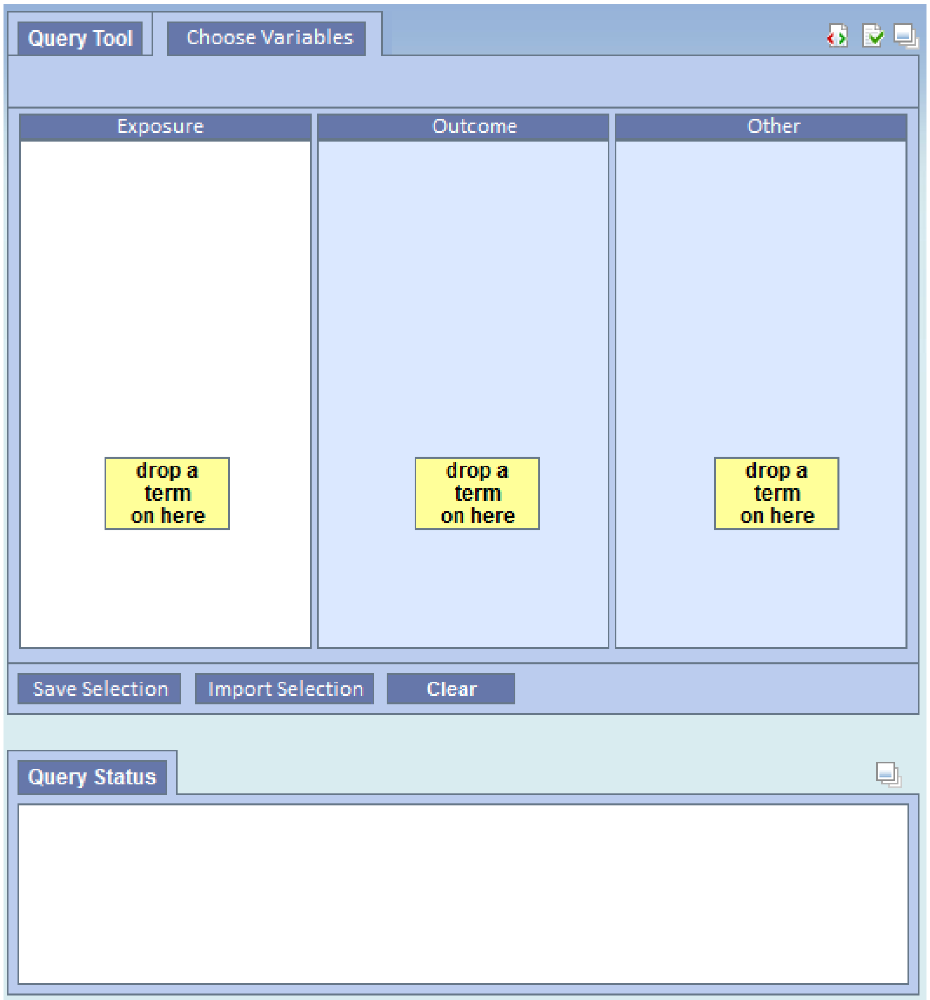


Figure 16: Planned new tab for the i2b2 webclient, allowing the selection of *Exposure*, *Outcome* and *Other* variables

In the new tab, there will be no i2b2 querying functionality. The three boxes act solely as a storage for the selected *Exposure*, *Outcome* and *Other* variables. The selected variables are planned to be exported and imported through the newly added buttons below.

4.2.3 Implemented customization of the i2b2 webclient

As a new feature, there is a new tab in the i2b2 webclient next to the tab of the query tool. It shows almost the same view as the original query tool with three new boxes to select the details of the researcher's query.

These boxes are entitled *Exposition*, *Outcome* and *Other*. In the *Exposition* box, the researcher is able to select all kinds of variables that the patient is exposed to, e.g. smoking daily. The *Outcome* box allows the specification of a patient's outcome, e.g. cardiovascular disease.

The last box called "Other" can be used for different variables that are neither outcome nor exposition, but which are still important for the study. That could e.g. be confounding variables. The *Other* box may also be used to simply *select* an item, not specifying that it is considered an exposition or an outcome.

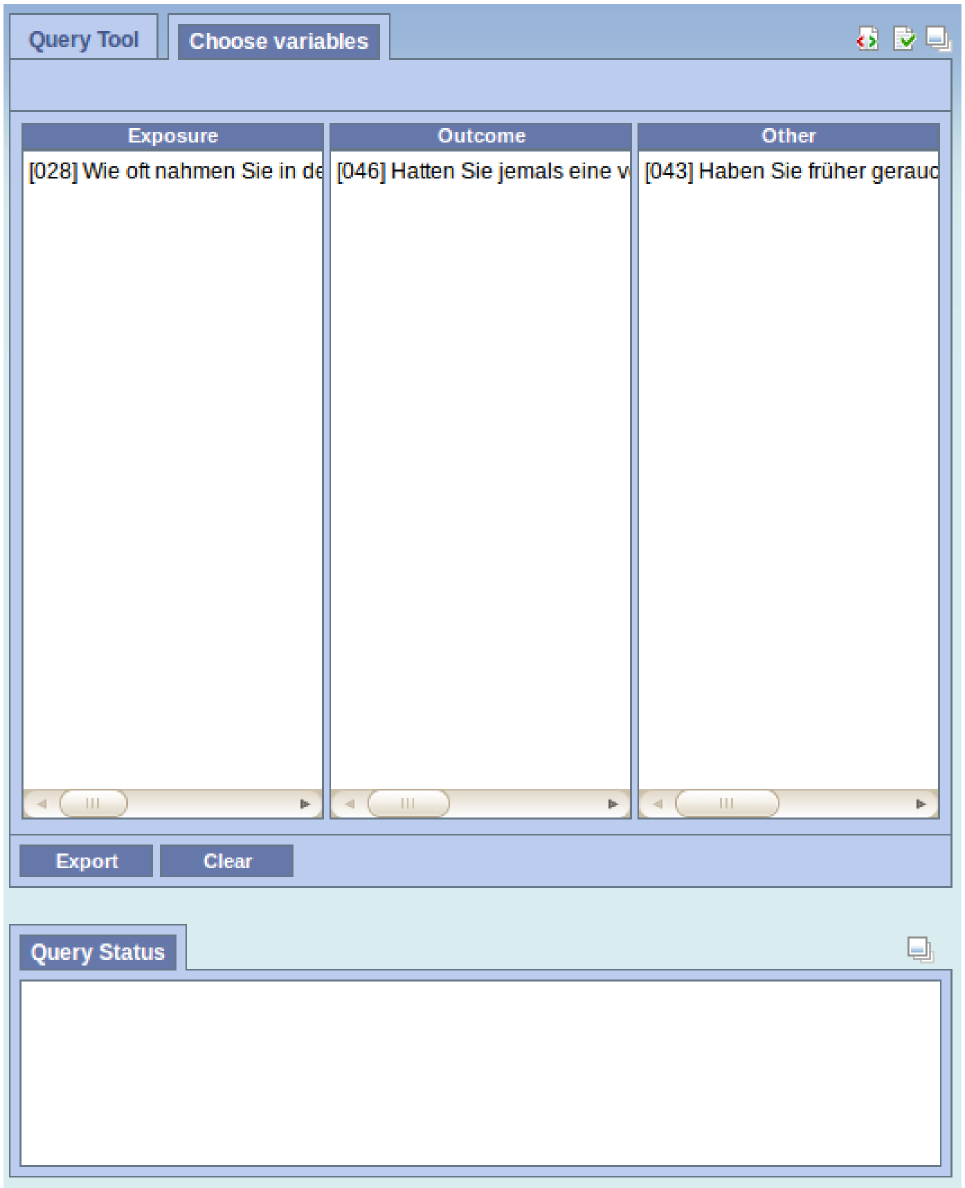
The whole process of selecting *Exposition*, *Outcome* and *Other* variables follows the i2b2-specific drag and drop functionality. It is therefore of an intuitive nature.

4.2.4 Implementation of the *Choose variables* tab

The newly created tab in the i2b2 webclient called *Choose variables* is partly a copy of the original i2b2 webclient's query tool, but with an altered functionality. The new tab consists of three panels, like the query tool. Below the panels are two buttons that allow an export of the selected items, as well as the deletion of all three boxes to start over with a new selection.

The deletion of a single item in that lists is possible through a context menu accessible through a right-click on the item. It is the same kind of context menu as in the i2b2 query tool.

The new tab looks as follows, containing three exemplary items dropped onto the three boxes:



Exposure	Outcome	Other
[028] Wie oft nahmen Sie in de	[046] Hatten Sie jemals eine v	[043] Haben Sie früher gerauc

Figure 17: New *Choose variables* tab with an example query

A researcher simply drags the items from the metadata tree into these three boxes. Compared to the previously shown mockup of the new tab, the developed tab contains only the buttons for exporting and clearing of the selection, which is sufficient for the scenario of importing the query to the transfer unit.

The *Import* functionality into the i2b2 webclient is not implemented. That would require the parsing of a local file on the client's side. This is a security threat to the client using JavaScript. JavaScript doesn't allow accessing local client side files. In a future version, a reimport of the Query could be accomplished, e.g. using PHP. Yet this solution may pose a security threat to the server that needs to be considered, too.

The implemented solution orientates itself at the i2b2-specific drag and drop functionality. Metadata items can be dragged from the metadata tree to the three newly added boxes in the webclient.

4.2.5 *Choose variables* tab data model

In order to store the items that are dropped onto the boxes *Exposure*, *Outcome* or *Other*, there is an internal client side JavaScript storage implemented for that purpose. The query tool on the one hand has its own storage system for managing the query items.

The three new boxes have their own internal i2b2 *panel_controllers* which handle the drag and drop functionality. Additionally, there are three client side JavaScript arrays set up which contain simple JavaScript objects that consist of a *name*, a *basecode* and a dimcode string. The *name* is the i2b2 C_NAME from the ontology table, which acts as the label shown to the user in the webclient. The *basecode* is the *C_BASECODE* from the i2b2 database. The dimcode is the C_DIMCODE from the ontology table, which contains the hierarchical information of the item.

The names of the three arrays are *arrayQPDChooseVariablesExposure*, *arrayQPDChooseVariablesOutcome* and *arrayQPDChooseVariablesOther*, which shows that each array stores the items of the box that is contained in the array name.

The idea of this approach is, that it is sufficient to contain these three values for each item to do the following:

- Store the items internally on the client side
- Display the items in the new *Choose variables* boxes to the user
- Exchange the *Exposure*, *Outcome* and *Other* items with the transfer unit

Once an item is dropped on one of the boxes, the server sends the client a message containing an XML formatted, so called *OrigData* block. This XML data contains the C_BASECODE, C_NAME and C_DIMCODE for the dropped metadata item.

Even though the three lists of metadata items may be enough for an exchange with the transfer unit, the inclusion and exclusion criteria should also be exchanged by using a known, standardized format.

4.2.6 Data export functionality

In order to get the selected items out of the i2b2 webclient and into the transfer unit, there needs to be an exporting functionality from the i2b2 webclient. The query data needs to be exported and reimported into the transfer unit's web application.

The export should work reliably and if possible by using a machine readable format. The mere text file generation and download functionality is achieved by using the utility *FileSaver.js*, which greatly simplifies the client side generation of a textfile [41].

By clicking the *Export* button on the webclient, the webclient generates a file on the user's computer. After that, there will be a download prompt opened by the webbrowser to let the user download that file to his computer. Later, the user will be able to log into the transfer unit's webapplication and upload that file to import his query there.

The query export data is split into two main parts, firstly the original i2b2 query tool metadata, and secondly the new *Choose variables* tab data. These two parts should both be exported to the transfer unit, which is achieved by generating an XML file that contains the necessary items.

The metadata in the query tool contains a variety of different logical constraints by which the metadata can be linked to each other. Each of the query tool boxes is a list of metadata items, which are related to each other in terms of an *OR* relation. The different boxes relate to each other with the logical *AND* relation.

Each box may have the following constraints:

- Be tagged as being *excluded*
- Have a temporal constraint like *Date from* and *Date until*

- Specify, how often these metadata items should occur (e.g. all patients whose blood pressure was elevated for at least 3 times)
- Treat different boxes independently, or that they should occur in the same patient encounter

4.2.7 Query tool metadata export

The i2b2 webclient stores the query tool selection internally on the client side for the dropped items. The webclient's data storage is read when submitting a query to the server application with the button *Run Query* being pressed. The webclient is then running the JavaScript function *_getQueryXML()* locally, which generates an XML string, representing the query tool content. It then sends that string to the application server, where it is reinterpreted and run against the database.

The XML string contains almost all the necessary information for the export of the inclusion and exclusion criteria to the transfer unit. Additionally to the already delivered XML data, there needs to be the *dimcode* and the *basecode* exported. Those contain the hierarchical information of the selected metadata items and also the *public_ddref_id* for each item, which is the GANI_MED specific data dictionary reference of that item. These two additional items are needed in order to import the query into the transfer unit.

In technical terms, the given *_getQueryXML* function has simply been copied and renamed to *_getQueryXMLExportGreifswald()* and was slightly customized for the needs of the query export. For customization, there is the need of adding the two new elements to the XML. Also, there are several tabs added for each line in the file in order to make the XML more human readable, if the file is to be read by a person.

4.2.8 *Choose variables* metadata export

The export of the *Choose variables* tab is very simple, because there is no logical constraint between or within the *Exposure*, *Outcome* and *Other* boxes. These three boxes act solely as a storage of three metadata item lists and the order of the items is of no meaning either. To export the selected items, the webclient reads the three object arrays and generates an XML string containing three lists of metadata items, each containing a public_ddref_id and the item's C_DIMCODE.

4.3 Example query

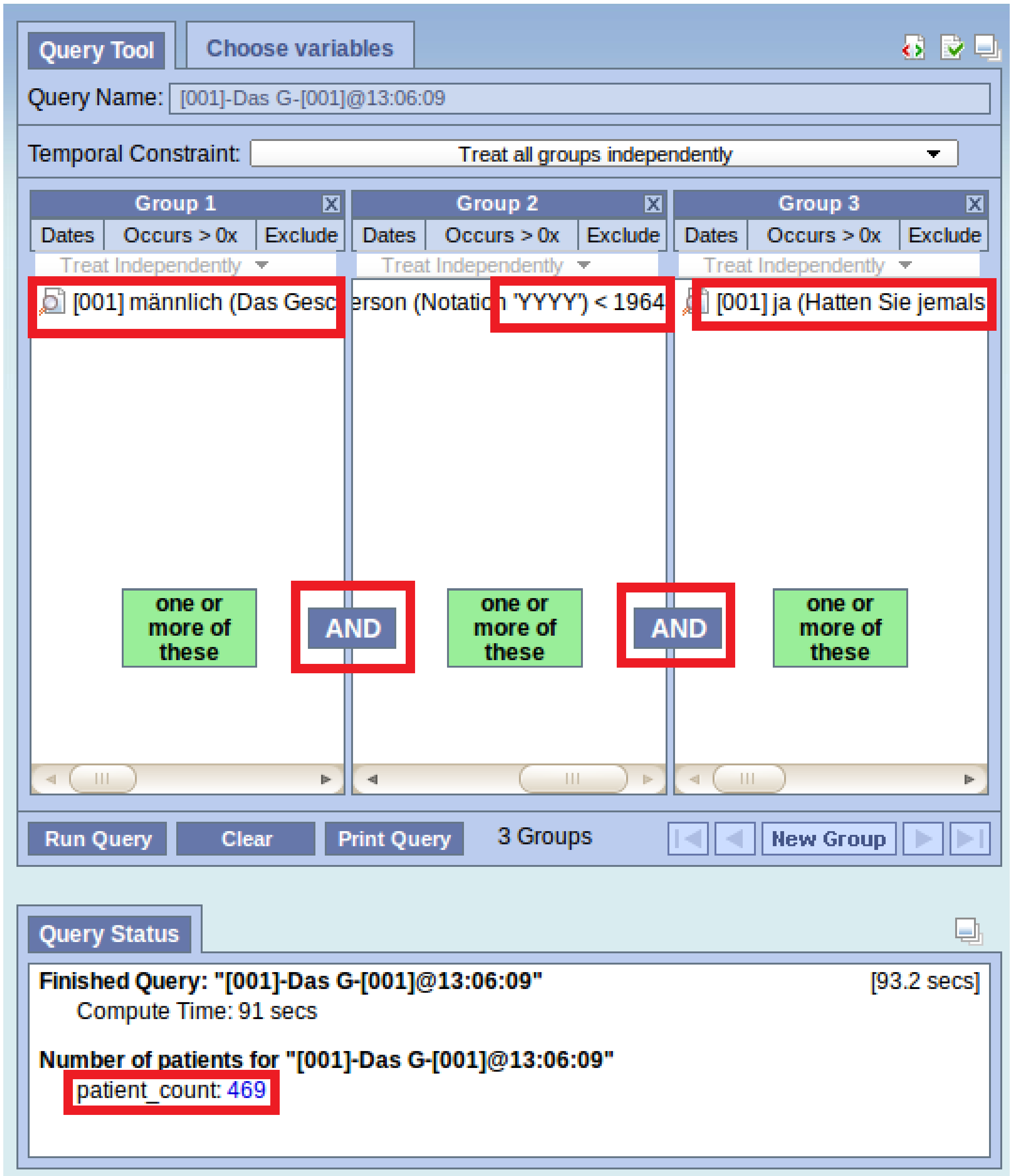
In this section, an exemplary query using i2b2 is performed and explained.

4.3.1 Defining inclusion and exclusion criteria

A researcher may be interested in data about the following patients:

> "All *male* patients, who are *older than 50* and who have had a *diagnosis of Angina pectoris* once before."

In i2b2, this query would be implemented as follows (Figure 18):



Query Name: [001]-Das G-[001]@13:06:09

Temporal Constraint: Treat all groups independently

Group 1: [001] männlich (Das Gesc

Group 2: erson (Notatio 'YYYY') < 1964

Group 3: [001] ja (Hatten Sie jemals

Finished Query: "[001]-Das G-[001]@13:06:09" [93.2 secs]

Compute Time: 91 secs

Number of patients for "[001]-Das G-[001]@13:06:09"

patient_count: 469

Figure 18: Querying for inclusion and exclusion criteria in the i2b2 webclient.

The red boxes indicate the before mentioned query for inclusion and exclusion criteria and the resulting patient count. The full names of the metadata items in the boxes are listed below:

- *Male* patients: "[001] männlich (Das Geschlecht einer Person ('F' steht für female, 'M' steht für male).)" (eng.: "[001] male (The gender of a person ('F' stands for female, 'M' stands for male)".)
- Patients being *older than 50*: "*Das Geburtsjahr einer Person (Notation 'YYYY') <1964*" (eng.: "The birthyear of a person (Notation 'YYYY') <1964")
- Patients who had a *diagnosis of Angina pectoris*: "[001] ja (Hatten Sie jemals eine von einem Arzt festgestellte Angina pectoris?)" (eng.: "[001] yes (Have you ever had a diagnosis of Angina pectoris, which was diagnosed by a physician?)")

The resulting patient count for that query is 469 patients.

4.3.2 Set up query for the transfer unit

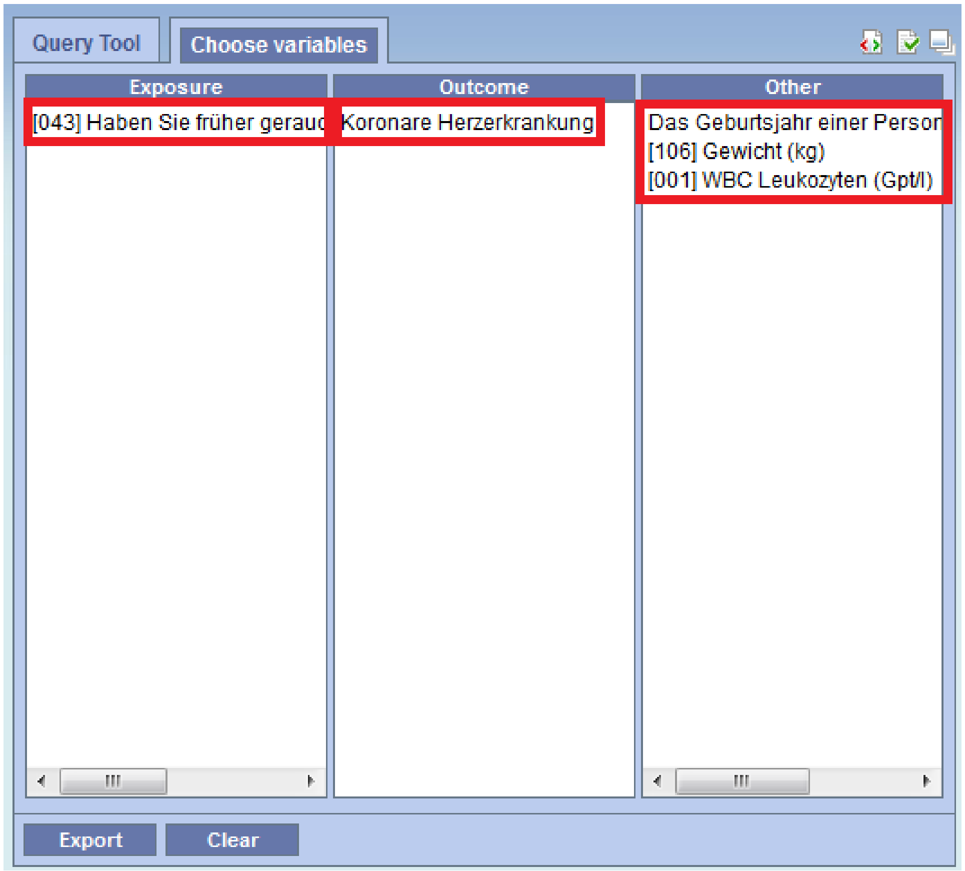
If the before mentioned patient count is large enough in size for the researchers needs, the following exemplary metadata items could be selected in the *Choose variables* tab (Table 14):

Item	**Chosen**	**Exposure**	**Outcome**
Body weight	x		
Birthyear	x		
Smoking status		x	
Coronary heart disease			x
White blood cell count	x		

Table 14: Example query, showing the *Exposure*, *Outcome* and *Other* categories.

In the before shown table, the items *age, body weight* and *white blood cell count* are *chosen* variables, whereas *smoking status* and *cardiovascular disease* are exposure and outcome variables, respectively.

In i2b2, these items would be selected in the *Choose variables* tab as follows (Figure 19):



Exposure	Outcome	Other
[043] Haben Sie früher gerauc	Koronare Herzerkrankung	Das Geburtsjahr einer Person
		[106] Gewicht (kg)
		[001] WBC Leukozyten (Gpt/l)

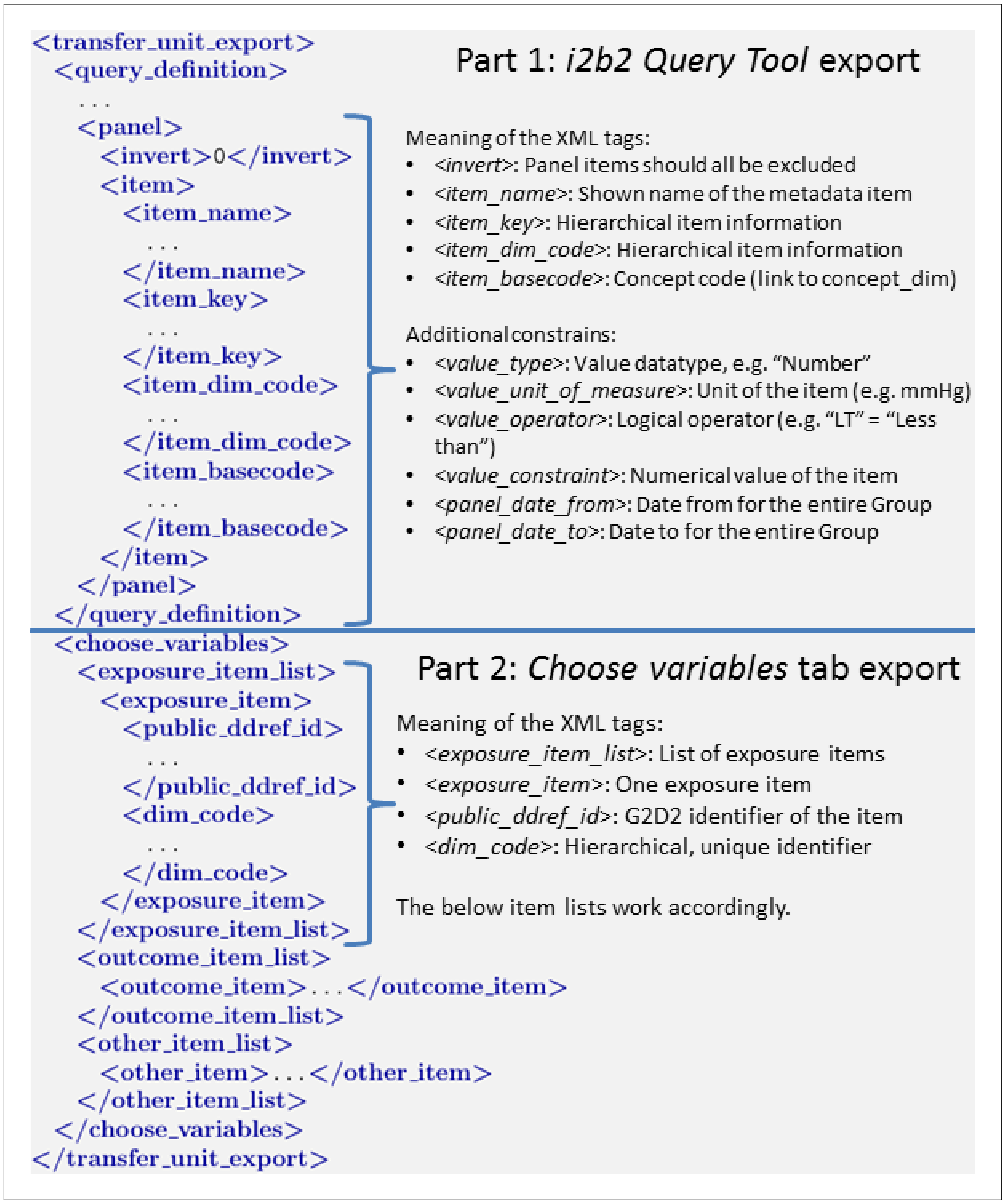
Figure 19: Exemplary selection of *Exposure*, *Outcome* and *Other* variables.

The full names of the selected metadata items in the *Choose variables* tab are:

- *Exposure*: "[043] Haben Sie früher geraucht oder rauchen Sie zurzeit?" (eng.: "Did you smoke before or are you currently smoking?")
- *Outcome*: "Koronare Herzerkrankung" (eng.: "Coronary heart disease")
- *Other*: "Das Geburtsjahr einer Person (Notation 'YYYY')" (eng.: "The birthyear of a person (Notation 'YYYY')")
- *Other*: "[106] Gewicht (kg)" (eng.: "[106] Weight (kg)")
- *Other*: "[001] WBC Leukozyten (Gpt/l)" (eng.: "WBC Leukocytes (Gpt/l)")

This query may then be exported using the *Export* button at the bottom left of the *Choose variables* tab.

The exported query file that will be uploaded to the transfer unit is shown below (Figure 20). Much information was skipped from the file below, the original can be found in the appendix (Listing 2):



Part 1: *i2b2 Query Tool* export

```
<transfer_unit_export>
  <query_definition>
    ...
    <panel>
      <invert>0</invert>
      <item>
        <item_name>
          ...
        </item_name>
        <item_key>
          ...
        </item_key>
        <item_dim_code>
          ...
        </item_dim_code>
        <item_basecode>
          ...
        </item_basecode>
      </item>
    </panel>
  </query_definition>
```

Meaning of the XML tags:

- <*invert*>: Panel items should all be excluded
- <*item_name*>: Shown name of the metadata item
- <*item_key*>: Hierarchical item information
- <*item_dim_code*>: Hierarchical item information
- <*item_basecode*>: Concept code (link to concept_dim)

Additional constrains:

- <*value_type*>: Value datatype, e.g. "Number"
- <*value_unit_of_measure*>: Unit of the item (e.g. mmHg)
- <*value_operator*>: Logical operator (e.g. "LT" = "Less than")
- <*value_constraint*>: Numerical value of the item
- <*panel_date_from*>: Date from for the entire Group
- <*panel_date_to*>: Date to for the entire Group

Part 2: *Choose variables* tab export

```
  <choose_variables>
    <exposure_item_list>
      <exposure_item>
        <public_ddref_id>
          ...
        </public_ddref_id>
        <dim_code>
          ...
        </dim_code>
      </exposure_item>
    </exposure_item_list>
    <outcome_item_list>
      <outcome_item>...</outcome_item>
    </outcome_item_list>
    <other_item_list>
      <other_item>...</other_item>
    </other_item_list>
  </choose_variables>
</transfer_unit_export>
```

Meaning of the XML tags:

- <*exposure_item_list*>: List of exposure items
- <*exposure_item*>: One exposure item
- <*public_ddref_id*>: G2D2 identifier of the item
- <*dim_code*>: Hierarchical, unique identifier

The below item lists work accordingly.

Figure 20: Exemplary exported query with reduced content.

5 Discussion

In this section, the results of this thesis are discussed. The before mentioned goals are evaluated, whether they were reached or not. The results are viewed from perspectives of effort, maintainability, as well as data security considerations. A discussion of this work and the related research literature is presented.

5.1 Thesis goals

This section discusses the before mentioned goals of this thesis, and whether they were reached or not.

5.1.1 First goal: Integration of metadata and research data into i2b2

The first goal of integrating the G2D2 metadata and the research data into i2b2 has been reached. It is now possible, to import the G2D2 metadata together with the research data into i2b2. Changes in the metadata or observational are possible and do not affect the data import, as long as the metadata and observational data follow the structure of G2D2 and the research platform database schema. Further metadata items and observations can simply be added and imported afterwards.

5.1.2 Second goal: Integration of the i2b2 webclient query

The second goal of integrating the i2b2 webclient query with the transfer unit was partly reached. The solution allows to export the query from the i2b2 webclient into a text file. This file contains the inclusion and exclusion criteria, as well as the detailed selection of variables for the transfer unit. It is still necessary, to develop a routine to reimport this file into the web application of the transfer unit. This development is not in the scope of this thesis and is to be solved by future works.

5.2 Effort and complexity of the solution

This section evaluates the effort that was necessary to solve the problem, as well as the complexity of the presented solution.

5.2.1 Data import into i2b2

The i2b2 data import routine was developed as a part of the G2D2 object model, namely as an *outcoming module*. This module is highly specific for both G2D2 and i2b2 and is therefore a tailored solution that cannot be seen independently of G2D2.

The effort of developing the i2b2 data import can be seen as moderate. The other, existing parts of G2D2 were already in a well developed state and didn't need reprogramming. The used metadata files went through several steps of customization, which were due to i2b2 specific needs. It was necessary to add i2b2 specific *AttributeTypeDefs* to the metadata. These were e.g. a specific *i2b2.ontology.name* acting as a label shown to the user in the webclient. Another example is an *AttributeTypeDef* called *i2b2.ontology.visible* to switch the visibility of single metadata items or entire branches on or off.

An excerpt of the Swisslab metadata containing the mentioned *AttributeTypeDefs* can be found in the appendix (Listing 3).

The overall complexity of the *OutI2B2SQLConnector* is moderate. The module utilizes a recursive method to read the G2D2 DTO model and generates SQL statements internally. The recursive part is the classical way of iterating over a hierarchy, where the i2b2-specific SQL generating part is merged into. This approach is therefore neither overly complex, nor simple.

5.2.2 Customized i2b2 webclient

The customized i2b2 webclient is a tailored solution for this project. The newly added tab *Choose variables* is developed rather specific for the transfer unit.

The division of metadata items into three categories, namely *Exposure*, *Outcome* and *Other* may also be used in different, epidemiological environments. It allows researchers to specify their query more through the selection of items in categories, rather than explaining their intentions in e.g. a free text form.

The customization of the webclient was chosen in order to provide the researcher with an additional feature that is similar to the existing *Query Tool*. The possibility of having another tab that looks almost like the

Query Tool was considered an elegant solution. This solution follows the i2b2 specific drag and drop functionality and was intended to be intuitive and easy to understand.

The method of customizing the i2b2 webclient was chosen in order to allow an intuitive use of the new functionality, without needing much adaptation of the user. Users who know the default i2b2 functionality of the Query Tool can rather easily learn how to use the newly introduced *Choose variables* tab. Also, the three newly added boxes act as an overview of already selected items. Hiding the already selected items in e.g. a context menu is not considered feasible.

The only thing a user needs to know is, what the three boxes mean in the view of the transfer unit. The user needs an introduction, which informs about the meaning of the *Exposure*, *Outcome* and *Other* boxes.

5.3 Maintainability

In order to maintain the implemented *OutI2B2SQLConnector* and the customized webclient, a developer needs to be familiar with both, G2D2 and JavaScript.

For the connector, the maintainability is good, because the connector is only one part of the G2D2 environment. It covers its own, encapsulated and specific functionality and bases itself on G2D2 input. Also, it is similar in functionality to other *outcoming modules* of G2D2, such as the *OutI2B2SQLWriter*.

For the customized webclient, the maintainability may be lower than for the *OutI2B2SQLConnector*. The reason for this is the highly tailored solution of altering the code of the original i2b2 webclient. The main part of the newly introduced functionality is covered in an additional JavaScript file. Yet, there are also necessary changes in the webclient source code in multiple other files, that belong to the original webclient. This approach renders the solution less maintainable by developers. For the user, having a new tab next to the *Query Tool* that also looks almost exactly the same as the *Query Tool*, may be convenient, easy to understand and easy to use. For a developer, this solution poses additional complication for further development. The solution for this problem should be further development of the webclient, allowing to show plug-ins as tabs next to the *Query Tool*.

The functionality of the *Choose variables* tab may be encapsulated in an i2b2 webclient plug-in.

5.4 Suitability of the applied tools

The applied tools used in this thesis were suitable for this solution. The developments using Eclipse and Aptana Studio were straightforward and easy to accomplish. Using Filesaver.js as a tool for exporting the i2b2 query was an elegant solution, regarding the simplicity and security of the file generation.

i2b2 itself is suitable for being used as a secondary data warehouse. It was possible, to import metadata and observational data into the i2b2 database and to query that data later on.

5.5 Independence from changing metadata or observational data

The implemented data import solution allows to alter the metadata and observational data and doesn't depend on a fixed set of items. Instead, the solution depends on a given structure of the data, namely the G2D2 object model and the database schema of the research database. It is possible to add or remove an arbitrary amount of metadata items from G2D2 as well as observations from the research database. After new metadata and / or observations are added, the import process would set up a new i2b2 database that contains the newly added items and their observations.

It is possible to include e.g. an entire new source information system as new metadata. The observations from the research database that belong to that new information system are then matched to the metadata during the import process.

In contrast to other works, the independence from the underlying metadata items can be seen as a novel approach of solving the data integration problem faced in clinical environments. The independence from the metadata allows to further add source information systems to the primary and secondary data warehouse.

Additional features in the metadata, such as the added i2b2-specific *AttributeTypeDefs* (Listing 3) have to be interpreted by the import program. These attributes are therefore fixed and need to be implemented by the *OutI2B2-SQLConnector*. The import is independent from changes in these attributes, once they are implemented in the connector. Independence from changes in the metadata means, that additional metadata can be added, if the meaning of the structure of that metadata is understood by the connector. Newly invented attributes and their meaning would yet not be understood by the import routine per se.

5.6 Secondary data warehouse

The secondary data warehouse is a system that needs to be designed in such a way that it can best serve the researcher's demand. It should contain clean and necessary data only.

5.6.1 Reasons for using a secondary data warehouse

A secondary warehouse can be seen as a data mart that has been derived from the primary data warehouse. Deriving the data mart could mean that there are necessary transformations and data cleansing procedures done on the data. In the derived data mart, only the relevant data from the primary data warehouse needs to be stored. This may save storage space and increase overall performance.

One of the main reasons for using a secondary data warehouse is that the application that is used to query the data warehouse expects a certain database structure. The primary data warehouse simply may not have that structure. Instead of changing the program code of the secondary data warehouse's application, it may be easier to just create another data warehouse using a database schema that the application expects to find.

5.6.2 Attributes

The general attributes of the secondary data warehouse are:

- Uses its own ETL process and matches data from the primary data warehouse with G2D2
- Contains quality assured and clean data only
- Gets the possibility of adding another data cleansing step beside the quality assurance, to fit the need of the query application (e.g. fil-

tering observational values that are not necessary or desired for the researcher)

- Has its own database schema that differs from the primary data warehouse's database schema
- Offers the possibility of querying the database by using a graphical program, the i2b2 webclient
- Uses i2b2 as a server side application to query the contained database
- Allows to export a researcher's query for reimport into the transfer unit's web application
- The graphical application shows the metadata to the user in a hierarchical pattern
- Users need to have a web browser to access the application, no installations are necessary

5.6.3 Objective

The main objective of the secondary data warehouse is that it should act as a primary contact point for researchers that want to study the data contained in the primary data warehouse. The secondary data warehouse offers the researcher the possibility of getting an idea of the data contained in the research platform, before requesting study data from the transfer unit. The secondary data warehouse can be used to ask for how many patients exist for a specific set of criteria. This query is to be specified by researchers through a graphical application. In this scenario, the metadata acts as a way to tell a researcher what data can be queried for in general. The observational data linked to the metadata can tell, how many patients fit the metadata that has been selected.

The secondary data warehouse is therefore meant to remove the current black box like situation of the research platform. This means that a researcher blindly requests data, hoping for a sufficient patient count for the selected criteria. This approach allows to get the researcher closer to the available data. It allows to get an understanding of the existing data before requesting detailed datasets.

5.6.4 Requirements

The secondary data warehouse has the following requirements:

- A specific ETL process needs to be programmed to match metadata with observational data that will be loaded into the data warehouse's database
- The i2b2 webclient used as the query application needs to be customized to fit the transfer unit's requirements

5.6.5 Reasons for using i2b2 for this scenario

Over the past years, i2b2 related research can be increasingly found in the area of medical research (Figure 21, [1]).

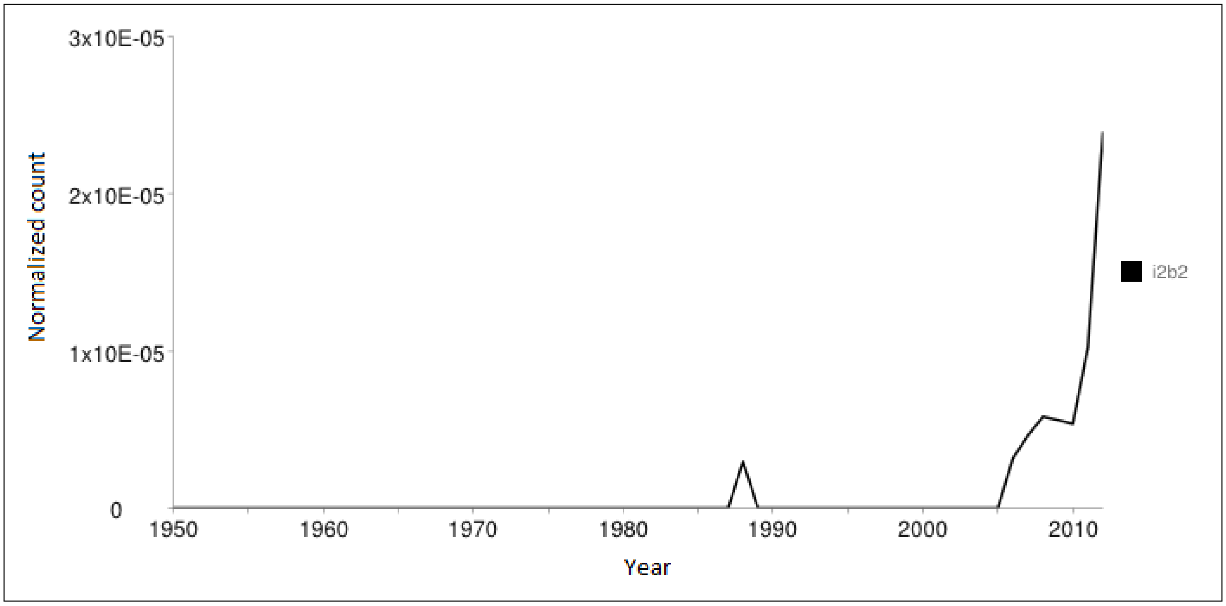


Figure 21: Term frequency of *i2b2* on Medline, using MLTrends search engine, by searching in publication's titles and abstracts, normalized over publication count [1].

Due to the fact that i2b2 offers the possibility of storing medical data that may as well come from different sources, it can principally be used as the secondary data warehouse.

i2b2-related research projects are found mainly in the North-American area, but there are also several research projects going on in the European area [42].

In Germany, the *Technologie- und Methodenplattform für die vernetzte medizinische For-schung e.V. (eng.: Technology and Methods Platform for Network Research in Medicine)* (TMF), a non profit organization dedicated to improve medical research is supportive of i2b2. Another argument is the active i2b2 user community [43]. Also, additional i2b2 functionality can be introduced relatively simple by adding so called i2b2 cells, which encapsulate functionality and are independent from each other. This may result in more new software features over time that may be easily added to the software.

The arguments of being suitable for this exact purpose, the increasing use and the active research community can be seen as the main reasons for using i2b2 for this scenario.

5.7 User management in i2b2

By using the i2b2 wizard, it is possible to add users to i2b2 that belong to the same project. By that technique, it is possible to create multiple users that use the same internal database for querying. Yet, at the transfer unit at Greifswald, there is a potentially large amount of users who have an account at the transfer unit. It would be inconvenient to add all these users additionally to the i2b2 project. It may be feasible, to add a functionality to i2b2 to allow a login using common authentication processes, such as the *Lightweight Directory Access Protocol* (LDAP). By using LDAP, it could be possible to authenticate each existing user in the transfer unit by using i2b2.

Otherwise, there need to be users in i2b2, which are used by multiple researchers. By using this approach, it would not be possible to identify each researcher in the i2b2 webclient, because other researchers may also use this i2b2 user account.

5.8 Data security measures and concerns

5.8.1 Data privacy

A potential threat to the whole i2b2 data warehouse strategy would be a user that could somehow take over the server and dump the entire i2b2 database. He could try to figure out, which patients are stored inside that database. The PATIENT_DIMENSION table contains only an auto incremented integer value as the patient's number (field *patient_num*) and the PATIENT_MAPPING table contains only the *patient_ide* value, a previously pseudonymized PID from the research database. Therefore, this potential threat may be rendered rather insignificant. Also, the ENCOUNTER_MAPPING table contains only a pseudonymized encounter ID from the research database.

What a potential hacker could figure out by capturing the database is, that there is e.g. a patient with certain laboratory results, who answered the GANI_MED specific questionnaire system *GANI_FORMS* in a specific way, and else. Yet, this knowledge may be of no use except for a clinician who actually remembers that patient, if he treated him. That is also true for anyone else knowing enough details about that patient to re-identify him.

There is no personal data stored in the i2b2 database, except the patient's gender and the birthyear which are very important for researchers. These items act as inclusion or exclusion criteria and are very valuable in many research scenarios.

What may increase the chance of finding out who an individual patient in the i2b2 database is, is the fact that the exact timestamp to the minute is contained in the OBSERVATION_FACT table. This is due to the data contained in the research database that holds this timestamp. That timestamp should be evaluated, whether it could be altered during the i2b2 import. It may be suitable to skip the time part and retain only the date part of the timestamp. For the i2b2 webclient's query tool, that would be sufficient to define a temporal constraint for the selected metadata items.

5.8.1.1 i2b2 patient count functionality

The i2b2 webclient allows a researcher to conveniently count the patient number for a given query using the i2b2 query tool. Since the result of that query is an exact number, a user may be able to figure out individual patients by querying the database in such a way, that a single patient may be returned.

For example, the user could create a query that is so detailed that there is only a single patient fitting that criteria, which may allow a physician to reidentify a patient in the database. That scenario is a scenario similar to the hacker that dumps the entire database, just by using the i2b2 webclient instead. Yet, it is much more complicated to create all that queries manually.

Both the hacker scenario and the here mentioned scenario should be evaluated by the responsible data protection officer, whether this is a significant threat or not. As a solution for the exact patient count scenario, the i2b2 webclient may just be altered in its functionality by returning intervals as a *patient count* result instead of an exact number. These intervals may just be using tens, saying there are less than 10 available patients, or between 10 and 20, etc.

5.8.1.2 i2b2 metadata tree at Greifswald

The metadata tree at the ICM is very detailed and allows very sophisticated queries on the underlying patient data. There should therefore be a data privacy evaluation of the i2b2 metadata tree, whether an i2b2 query may be conducted so sophistically, or in other words, whether that metadata tree is so detailed and precise, that it may be possible to identify a patient by his medical data, age and gender. That may for instance be done by a person that knows that patient personally, but didn't treat him or her.

An example may be a very rare disease visible to other people on the patient's outside body and which could be queried for in i2b2. Additionally, the query could then go for the patient's gender and age, to figure out, who that person could be. If then there is a user who figures out that this very person he knows has a drug addiction, a psychiatric diagnosis, or maybe an HIV infection, that would then turn out to be a social stigma for that person. That may in the end lead to unemployment of that patient, for instance.

These mentioned privacy related scenarios may sound sophisticated to occur or to achieve. Yet due to the fact that this database comprises of multiple source systems that can multiply the overall contained information by consolidating them, it needs to be treated with care. This is due to the fact, that for instance psychiatric and drug related diagnoses are stigmatizing patients, when they are known to others.

i2b2 should not be put into productive use without the approval of a data protection officer, who evaluates the Greifswald i2b2 metadata tree and the query tool. That person should also be informed about future changes in the detail of the metadata.

The consolidation of multiple source systems is a great improvement for researchers, allowing them to get a vast and very detailed view on a patient group that he is interested in. Yet, technological advancements may pose threats to data privacy that should not be underestimated and should be carefully evaluated. The need for research to improve healthcare and the chances of identifying individual patients and potentially threaten their social identity should be carefully balanced and be decided on. Yet there should be a decision in favor of the patient's data privacy, if the risk of reidentification is too high and cannot be accepted. This could mean to skip certain items from the metadata tree, which should be decided by an independent data protection officer.

5.8.2 File generation for query export

The file generator for the export of the i2b2 query is making use of a client side tool, using JavaScript and may therefore be of no risk to the i2b2 server. An alternative method would have been the usage of a PHP-based approach, where there is a PHP-script executed on the server. That script would then generate a server side textfile, which is then sent back to the client.

This approach has been evaluated, but it has been skipped due to security reasons. Since the client needs to send a string (the file's content) to the server in order to be able to generate that file, the client may just send a string of PHP programming code. That code may then be executed by the server's PHP engine, causing a server crash, or else. A client side file generation is a safer way to avoid this problem and is therefore used here.

5.9 Alternative to a customized webclient

In a future work, the tailored solution of a specifically customized i2b2 webclient may be altered. By including a way to specify biomaterial specimens, an i2b2 webclient plug-in may be developed. In that plug-in, a researcher may specify both, the lists of the metadata items that are needed, as well as what biomaterial may be requested. That solution may be similar in parts to the *Onco-i2b2 Biobank Info plug-in* [44].

An important note is, that this solution would again be tailored to a scenario which contains specifically the given categories of *Exposure*, *Outcome* and *Other* variables.

5.10 Further i2b2 webclient development

The i2b2 webclient is a convenient tool for researchers who are distributed over multiple institutions. The possibility of accessing the underlying data without the need of installing any software beside a web browser is very simple and feasible.

Future works could further develop the i2b2 webclient functionality. It would e.g. be useful to access i2b2 webclient plug-ins more directly. One way could be to let plug-ins optionally show up in the webclient as additional tabs next to the Query Tool tab. This approach makes plug-ins accessible faster. Today, the plug-ins have to be accessed through the *Analysis Tools* section instead. This makes the existing plug-ins hidden in the application, rather than visible. Adding an optional plug-in tab for each plug-in next to the *Query Tool* tab could be an elegant way of solving this problem.

5.11 Related works

This section deals with other works that are related to this thesis.

5.11.1 IDRT - Integrated Data Repository Toolkit

The IDRT project partly aimed at creating a way for importing metadata and observational data into i2b2. i2b2 doesn't offer its own ETL process in order to import data into the i2b2 database. The *IDRT Import Tool* allows to import data from different datasources, e.g. *Operational Data Model* (ODM) or *Comma-Separated Values* (CSV) files into i2b2 [45].

As an example, an ODM file may contain the metadata and observational data about a clinical study. That data would then be imported into i2b2 using the *IDRT Import Tool*.

The metadata in the GANI_MED project follows its own model, namely the G2D2 object model. The use of an independent metadata model was chosen, because G2D2 was considered as being a more suitable solution for this project.

The exact reasons and possible technical constrains for not using ODM may be evaluated in the future.

5.11.2 HL7 ETL cell

Another related project deals with an automated realtime data import for i2b2, conducted in 2012 at Gießen University, Germany. This project aims at creating a tool that is using realtime *Health Level Seven* (HL7) version 2 data for importing clinical data into i2b2 [46]. The realtime input data is taken directly from an integration server. Those servers are considered as being available in many hospitals. The resulting software *HIStream* is currently in development [47].

This solution has not been evaluated for this project. The fact that it relies solely on HL7 messages as data input could mean that it may not be usable in this context. Also, the developmental status of the *HIStream* software prevented an evaluation.

6 Outlook

In this section, future steps related to this work are described.

6.1 Key user evaluation of i2b2

Following this work, there should be an evaluation of i2b2 taking place from the view of a group of users. These users should be *key users*, meaning that they are real medical or epidemiological researchers that are going to use i2b2 in the future.

By using an evaluation, the following points may be adressed:

- Structure of the metadata tree shown in the webclient (being suitable for the researcher's needs)
- Ordering of the metadata items as being logical
- General, optical impression of i2b2
- Items as being quickly findable and categories as being complete
- Determining, which tasks i2b2 may take for a researcher (Not replacing the transfer unit)
- Determining the actual benefits for researchers, which are introduced by i2b2

6.2 Import of the researcher's query into the transfer unit's web application

In order to integrate i2b2 with the transfer unit, the query exported from the i2b2 webclient needs to be imported into the transfer unit's web application. This import routine needs to parse the exported i2b2 query file and set the selected metadata items in the transfer unit's web application.

Additionally to the mentioned *Exposure*, *Outcome* and *Other* variables, the inclusion and exclusion criteria from the i2b2 Query Tool may be imported as well.

6.3 Data privacy

Before i2b2 is offered to researchers, a data protection officer should look over the database and the webclient to check, whether the data privacy of patients is ensured. After an approval, i2b2 may be deployed and made available to researchers. If there are data privacy concerns, then i2b2 should be further customized and the database contents overworked, in order to meet the privacy needs.

Appendices

A Appendix

A.1 Screenshots

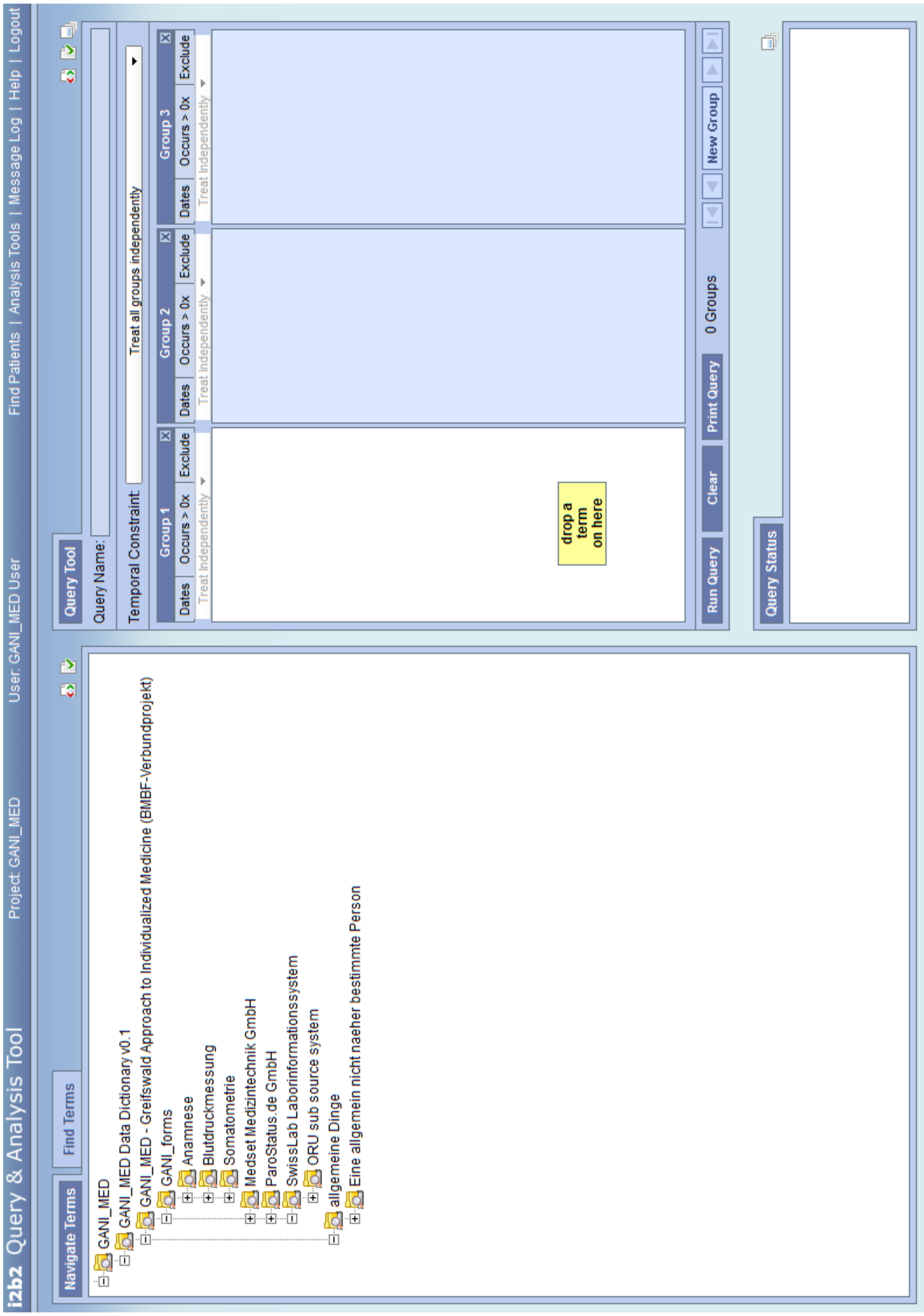


Figure 22: i2b2 webclient [48].

A.2 ETL process for loading the research database

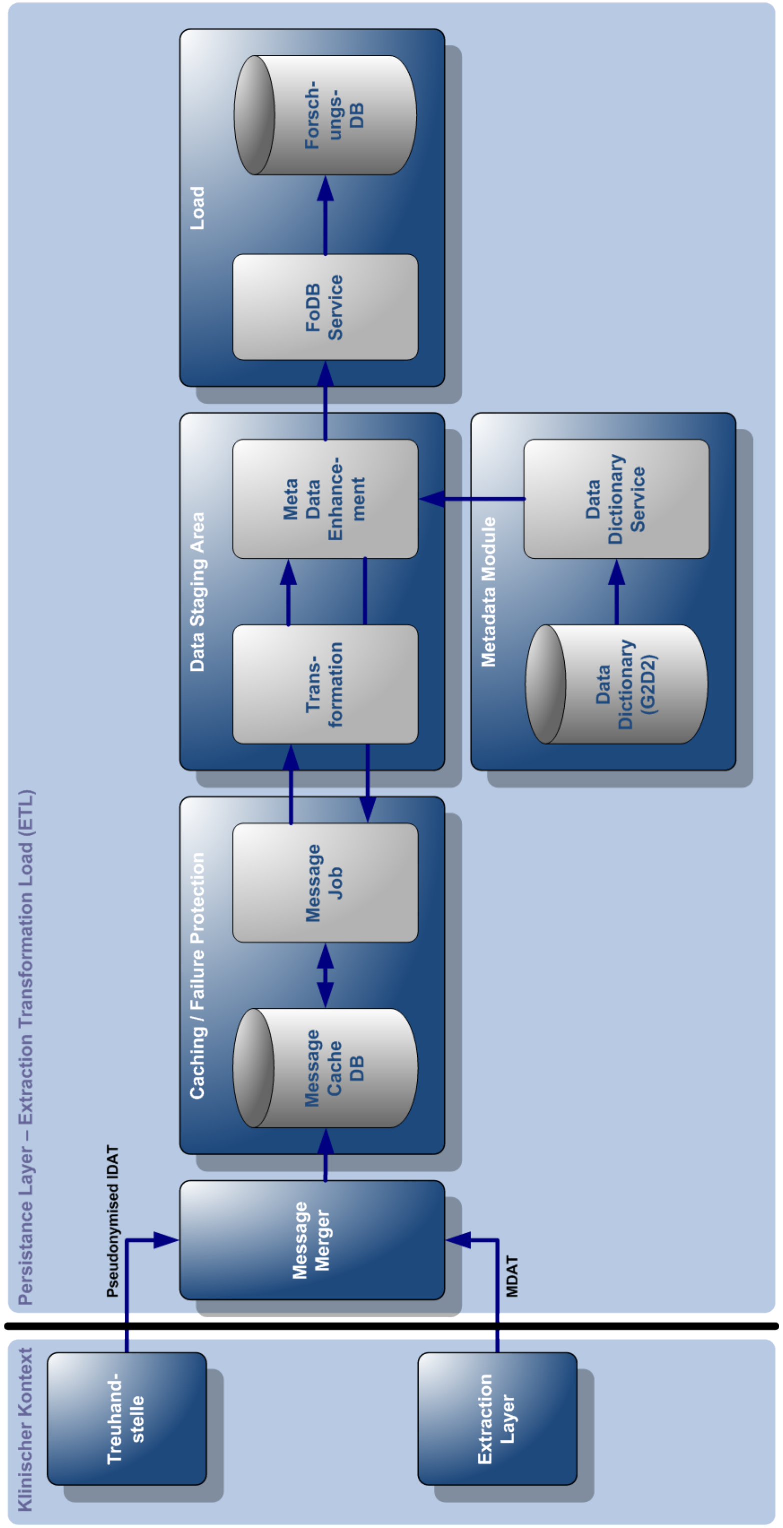


Figure 23: ETL process for loading the research database (Forschungs-DB), [49].

A.3 Process model of the research data application process

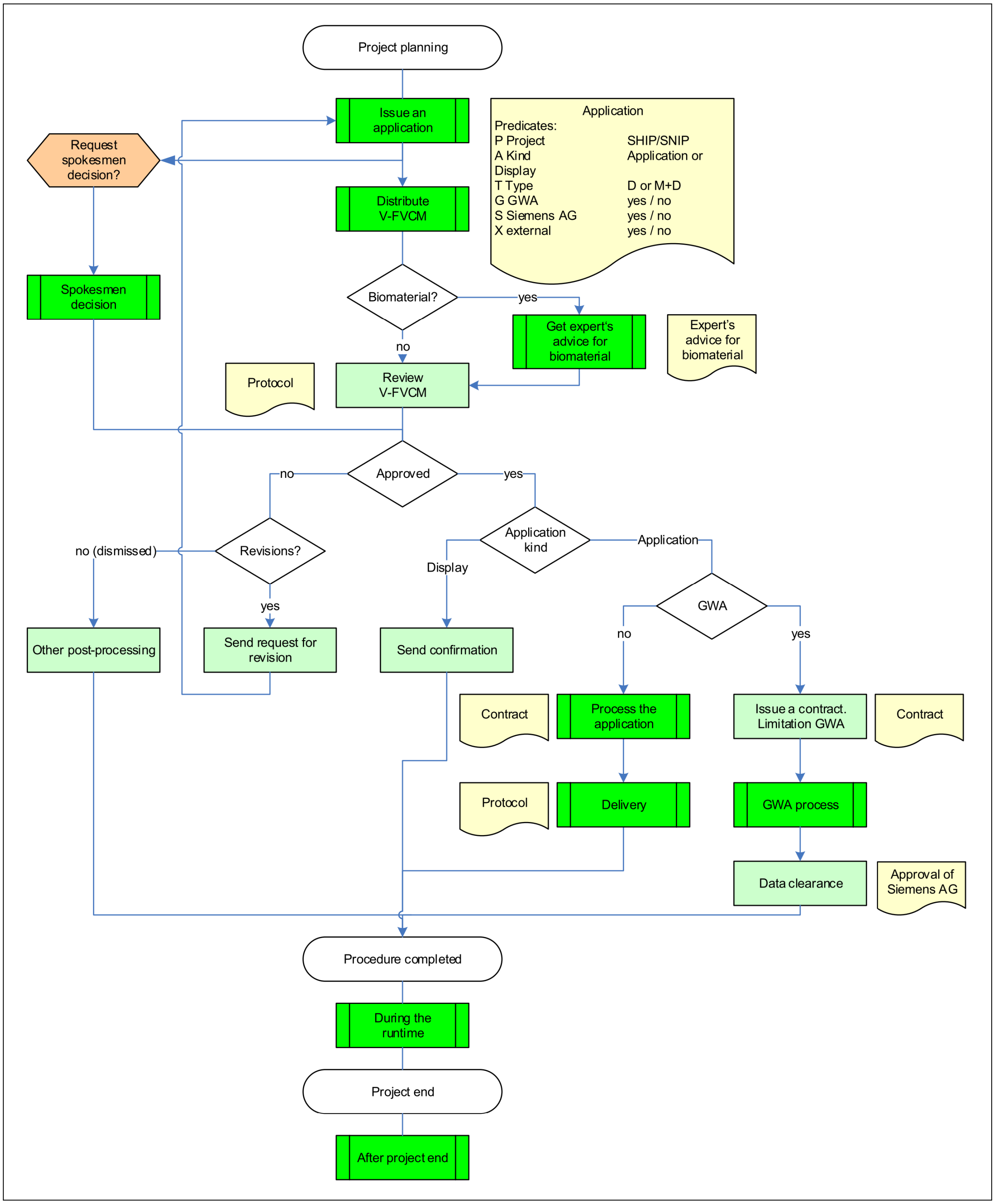


Figure 24: Detailed process model of the data application at the transfer unit [24].

A.4 BPMN model of the research data application process

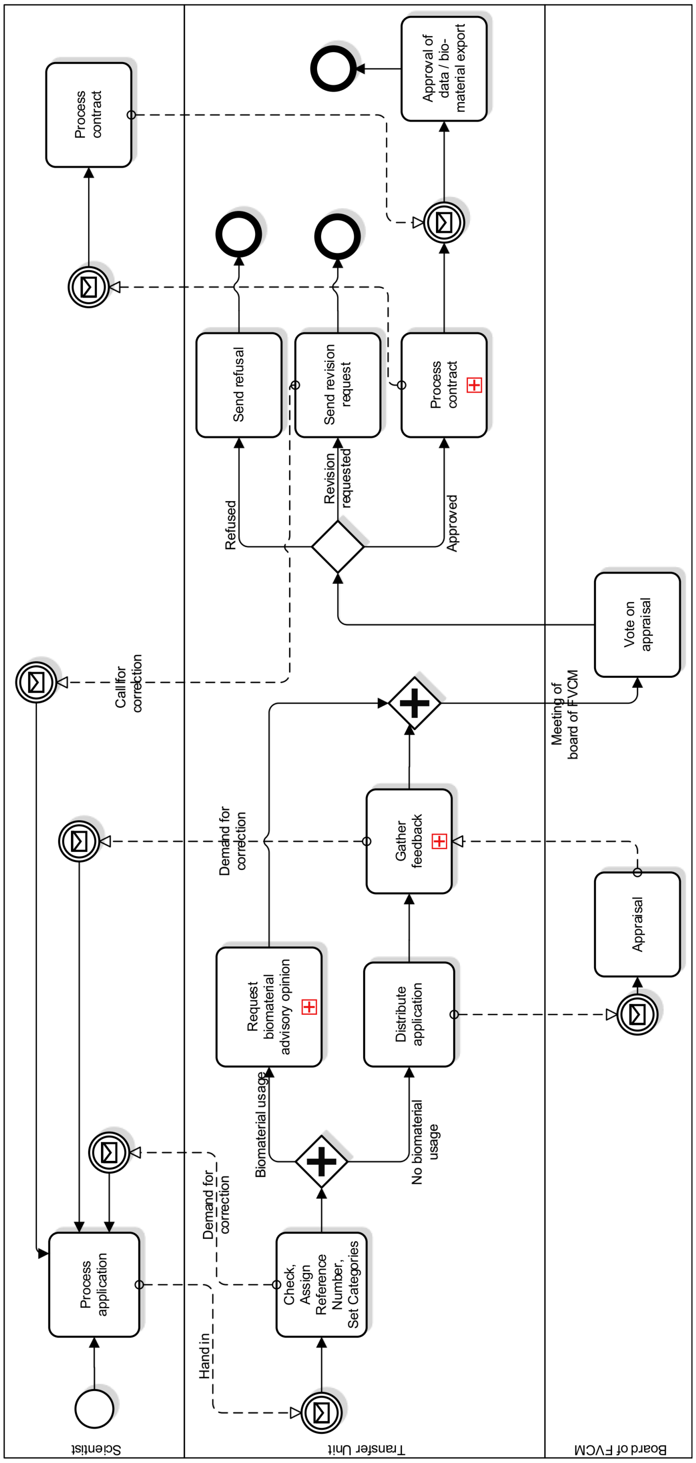


Figure 25: BPMN process of applying for research data at the transfer unit [23].

A.5 G2D2 DTO models

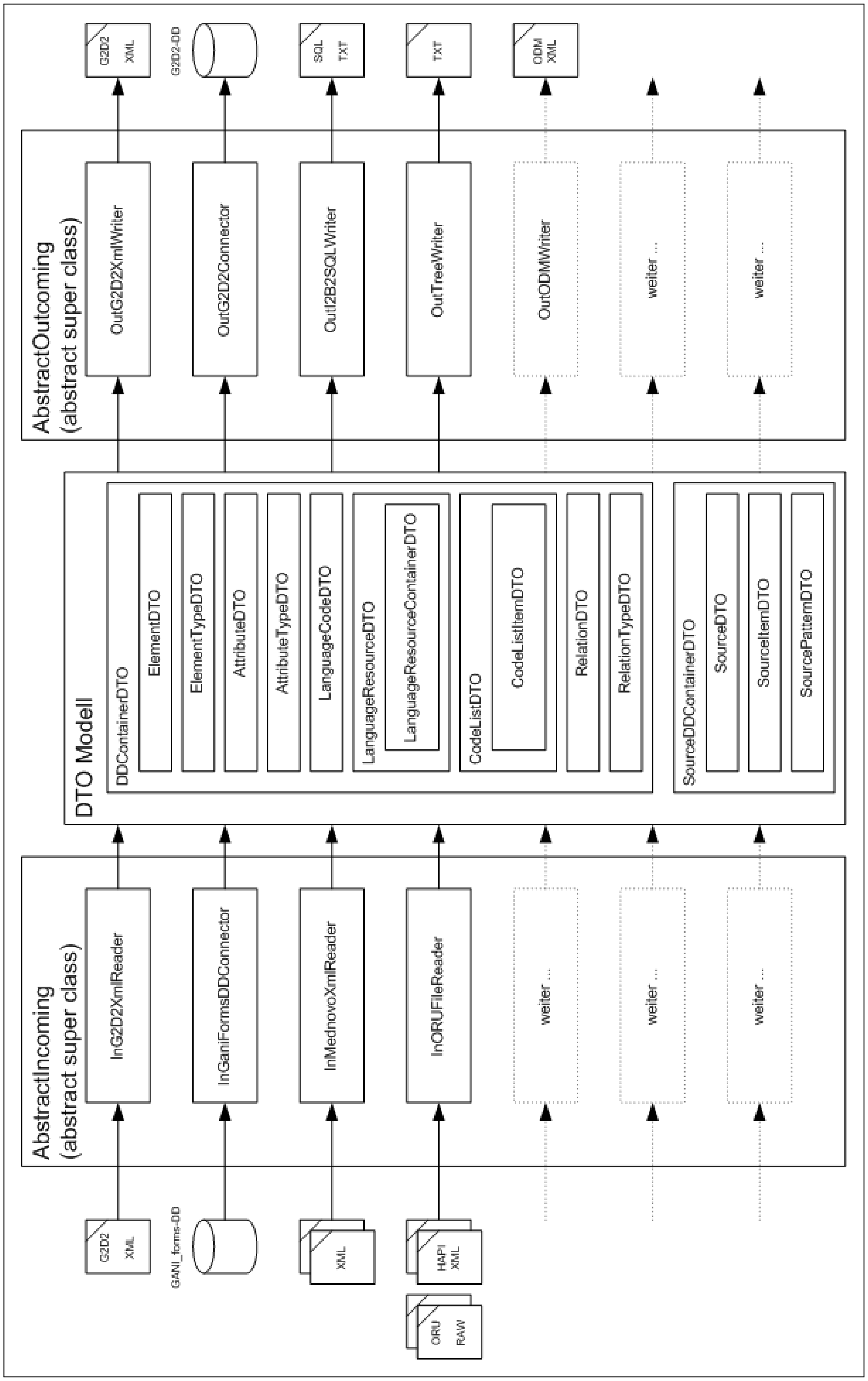


Figure 26: Metadata extraction and conversion strategy [25].

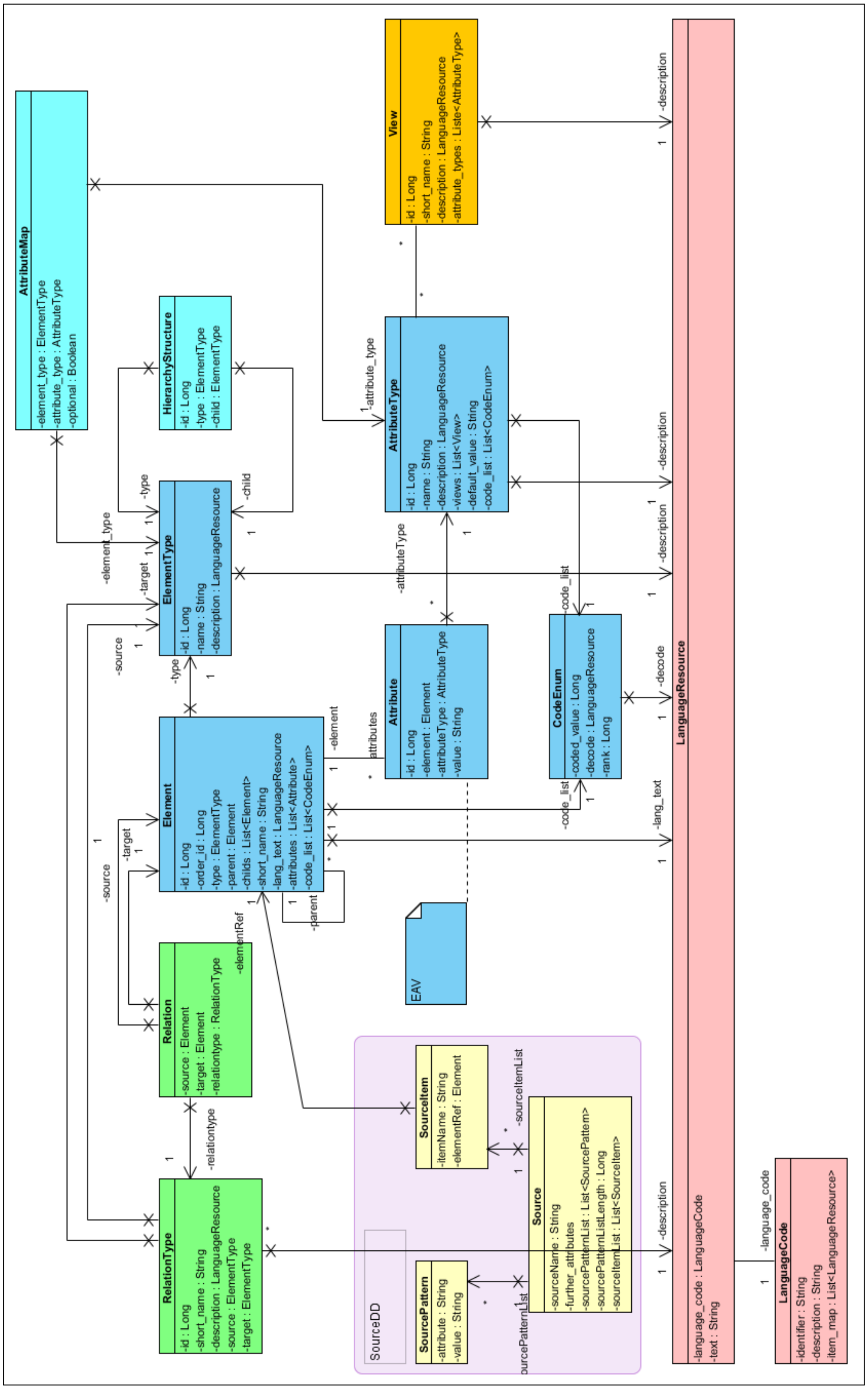


Figure 27: G2D2 DTO UML model [25].

A.6 XML: Example export query from i2b2 webclient

```
<transfer_unit_export>
  <query_definition>
    <query_name>Transfer_unit_export</query_name>
    <query_timing>ANY</query_timing>
    <specificity_scale>0</specificity_scale>
    <panel>
      <panel_number>1</panel_number>
      <panel_accuracy_scale>100</panel_accuracy_scale>
      <invert>0</invert>
      <panel_timing>ANY</panel_timing>
      <total_item_occurrences>1</total_item_occurrences>
      <item>
        <hlevel>6</hlevel>
        <item_name>[001] männlich (Das Geschlecht einer Person (’F’ steht für female, ’M’ steht für male).)</item_name>
        <item_key>\\GANI_MED\GANI_MED\GANI_MEDdatadictionary\common\person\patient\gender\M\</item_key>
        <item_dim_code>\GANI_MED\GANI_MEDdatadictionary\common\person\patient\gender\M\</item_dim_code>
        <item_basecode>common-n_src_vers-person.patient.gender:M</item_basecode>
        <tooltip>männlich</tooltip>
        <class>ENC</class>
        <item_icon>LAE</item_icon>
        <item_is_synonym>false</item_is_synonym>
      </item>
    </panel>
    <panel>
      <panel_number>2</panel_number>
      <panel_accuracy_scale>100</panel_accuracy_scale>
      <invert>0</invert>
      <panel_timing>ANY</panel_timing>
      <total_item_occurrences>1</total_item_occurrences>
      <item>
        <hlevel>5</hlevel>
        <item_name>Das Geburtsjahr einer Person (Notation ’YYYY’) &lt; 1964 </item_name>
        <item_key>\\GANI_MED\GANI_MED\GANI_MEDdatadictionary\common\person\patient\birthyear\</item_key>
        <item_dim_code>\GANI_MED\
          GANI_MEDdatadictionary\common\person\
          patient\birthyear\</item_dim_code>
        <item_basecode>common-n_src_vers-person.
          patient.birthyear</item_basecode>
        <tooltip>Das Geburtsjahr einer Person (Notation
          'YYYY')</tooltip>
        <class>ENC</class>
        <item_icon>LAE</item_icon>
        <item_is_synonym>false</item_is_synonym>
      <constrain_by_value>
        <value_type>NUMBER</value_type>
        <value_unit_of_measure> </value_unit_of_measure>
        <value_operator>LT</value_operator>
        <value_constraint>1964</value_constraint>
      </constrain_by_value>
      </item>
    </panel>
    <panel>
      <panel_number>3</panel_number>
      <panel_accuracy_scale>100</panel_accuracy_scale>
      <invert>0</invert>
      <panel_timing>ANY</panel_timing>
      <total_item_occurrences>1</total_item_occurrences>
      <item>
        <hlevel>7</hlevel>
        <item_name>[001] ja (Hatten Sie jemals eine von einem Arzt
          festgestellt Angina pectoris?)</item_name>
        <item_key>\\GANI_MED\GANI_MED\
          GANI_MEDdatadictionary\gani_med\gani_forms
          \anamnese\hkKardBeschw\hk_ang_pect\1\</item_key>
        <item_dim_code>\GANI_MED\ GANI_MEDdatadictionary\gani_med\
          gani_forms\anamnese\hkKardBeschw\ hk_ang_pect\1\
          </item_dim_code>
        <item_basecode>gani_forms.anamnese.
          hkKardBeschw-n_src_vers-hk_ang_pect-1:1 </item_basecode>
        <tooltip>ja</tooltip>
        <class>ENC</class>
        <item_icon>LAE</item_icon>
        <item_is_synonym>false</item_is_synonym>
      </item>
    </panel>
  </query_definition>
  <choose_variables>
    <exposure_item_list>
```

```
      <exposure_item>
        <public_ddref_id>gani_forms.anamnese.
           allgAnam-n_src_vers-rauch </public_ddref_id>
        <dim_code>\GANI_MED\GANI_MEDdatadictionary\
           gani_med\gani_forms\anamnese\allgAnam\rauch\ </dim_code>
      </exposure_item>
    </exposure_item_list>
    <outcome_item_list>
      <outcome_item>
        <public_ddref_id>undefined</public_ddref_id>
        <dim_code>\GANI_MED\GANI_MEDdatadictionary\
           gani_med\gani_forms\anamnese\hkKhk\ </dim_code>
      </outcome_item>
    </outcome_item_list>
    <other_item_list>
      <other_item>
        <public_ddref_id>common-n_src_vers-person.
           patient.birthyear</public_ddref_id>
        <dim_code>\GANI_MED\GANI_MEDdatadictionary\
           common\person\patient\birthyear\ </dim_code>
      </other_item>
      <other_item>
        <public_ddref_id>gani_forms.soma.soma-
           n_src_vers-soma_gew-1</public_ddref_id>
        <dim_code>\GANI_MED\GANI_MEDdatadictionary\
           gani_med\gani_forms\soma\soma\soma_gew\ </dim_code>
      </other_item>
      <other_item>
        <public_ddref_id>swisslab.ikch-n_src_vers-
           WBC,E-1</public_ddref_id>
        <dim_code>\GANI_MED\GANI_MEDdatadictionary\
           gani_med\swisslab\ikch\WBC,E\</dim_code>
      </other_item>
    </other_item_list>
  </choose_variables>
</transfer_unit_export>
```

Listing 2: Example export query from i2b2 webclient.

A.7 XML: Swisslab G2D2 metadata, i2b2 related AttributeTypeDefs

```
<AttributeTypeDefs>
        <AttributeTypeDef Name=”i2b2.ontology.visible” OID=”AT.4”>
            <DefaultValue>0</DefaultValue>
            <DescriptionLanguageResourceRef>LR.1375
                </DescriptionLanguageResourceRef>
            <CodeListRef>CL.3</CodeListRef>
        </AttributeTypeDef>
        <AttributeTypeDef Name=”i2b2.ontology.orderid” OID=”AT.5”>
            <DescriptionLanguageResourceRef>LR.1378
                </DescriptionLanguageResourceRef>
        </AttributeTypeDef>
        <AttributeTypeDef Name=”i2b2.ontology.name” OID=”AT.6”>
            <DescriptionLanguageResourceRef>LR.1379
                </DescriptionLanguageResourceRef>
        </AttributeTypeDef>
        <AttributeTypeDef Name=”i2b2.ontology.type” OID=”AT.7”>
            <DescriptionLanguageResourceRef>LR.1380
                </DescriptionLanguageResourceRef>
      </AttributeTypeDef>
</AttributeTypeDefs>
```

Listing 3: Excerpt of the Swisslab G2D2 metadata, showing i2b2-specific *AttributeTypeDefs*.

Acronyms

AUG

Academic User Group. 32

BPMN

Business Process Modelling Notation. 26, 85

CSV

Comma-Separated Values. 77

DTO

Data Transfer Object. 27, 32, 33, 44, 67, 87

EAV

Entity-Attribute-Value. 14, 18, 23, 25

ECG

Electrocardiography. 23

EHR

Electronic Health Record. 9

ETL

Extract, transform, load. 17, 21–23, 70, 72, 77, 83

G2D2

Generic GANI_MED Data Dictionary. 13, 14, 19, 23, 27, 32, 33, 36, 37, 40–44, 50–52, 54, 67–70, 78, 87, 91

GANI_MED

Greifswald Approach to Individualized Medicine. 5, 11, 13, 21, 22, 25, 27, 45, 60, 74, 78

GNU

GNU's Not Unix. 28

HL7

Health Level Seven. 78

i2b2

Informatics for Integrating Biology and the Bedside. 5, 13–15, 19, 23, 28–37, 39–41, 43–56, 58–62, 64, 66–69, 71–79, 82, 90, 91

ICM

Institute for Community Medicine. 5, 11, 32, 33, 75

IDE

Integrated Development Environment. 28

IDRT

Integrated Data Repository Toolkit. 32, 77, 78

LDAP

Lightweight Directory Access Protocol. 73

NIH

National Institutes of Health. 30

ODM

Operational Data Model. 77, 78

RCP

Rich Client Platform. 28, 30

RDBMS

Relational Database Management System. 18, 29

SOAP

Simple Object Access Protocol. 30

TMF

Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V. (eng.: Technology and Methods Platform for Network Research in Medicine). 73

UML

Unified Modeling Language. 30, 87

VM

Virtual Machine. 53

XML

Extensible Markup Language. 33, 42, 59–61

List of Figures

1 Term frequency of *Big data*, *Secondary use* and *Data warehouse* on Medline 13
2 Term frequency of terms related to individualized medicine and biological data on Medline 14
3 Entity-attribute-value format 22
4 Systematic overview ETL, primary and secondary data warehouse 26
5 General high level structure of the values, observations and entities of the research database 28
6 Exemplary high level structure of the values, observations and entities of the research database 28
7 transfer unit web application 29
8 Simplified BPMN data application process 30
9 G2D2 DTO model, simplified 31
10 i2b2 hive star schema 35
11 Import routine architecture 38
12 Mapping of the research database query with the i2b2 database 43
13 Mapping of G2D2 to the i2b2 ontology table 45
14 Importing the G2D2 metadata into the i2b2 ontology table . . 47
15 i2b2 data import method 53
16 i2b2 webclient new tab for exposure, outcome and other variables 59
17 New *Choose variables* tab with an example query 61
18 Querying for inclusion and exclusion criteria in i2b2 66
19 Exemplary selection of *Exposure*, *Outcome* and *Other* variables 68
20 Exemplary exported query, reduced information 69
21 Term frequency of *i2b2* on Medline 76
22 i2b2 webclient 86
23 ETL process to load research database 87
24 Process model of the data application 88
25 BPMN data application process 89
26 Metadata extraction and conversion strategy 90
27 G2D2 DTO UML model 91

List of Tables

1 Additional G2D2 attributes for i2b2 ontology items. 23
2 Additional G2D2 attributes for i2b2 ontology items. 37
3 i2b2 ontology table [40] 40
4 i2b2 CONCEPT_DIMENSION table [40] 41
5 Example observation from the research database (single row) . 42
6 G2D2 element definition for *ElementDef* 44
7 i2b2 PATIENT_DIMENSION table [40] 49
8 i2b2 PATIENT_MAPPING table [40] 50
9 i2b2 ENCOUNTER_MAPPING table [40] 51
10 i2b2 OBSERVATION_FACT table [40] 52
11 Hardware specification data import test 57
12 Performance test result of the data import, *Computer 1* 57
13 Performance test result of the data import, *Computer 2* 58
14 Example query, showing the *Exposure*, *Outcome* and *Other* categories 67

References

[1] Palidwor et al. (2010) J. Biomed. Discov. Collab. 5, 1-6 (Last accessed: Dec 19, 2013). `http://www.ogic.ca/mltrends/`.

[2] U.S. Department of Health & Human Services. Doctors and hospitals' use of health IT more than doubles since 2012 (Last accessed: Dec 19, 2013). `http://www.hhs.gov/news/press/2013pres/05/20130522a.html`.

[3] K. Pommerening. Secondary Use of the EHR via Pseudonymisation. *Studies in health technology and informatics*, 2004.

[4] Cynthia A. Brandt et al. Secondary Use of EHR: Data Quality Issues and Informatics Opportunities. *AMIA Summits Transl Sci Proc*, 2010.

[5] David C. Whitcomb. What is personalized medicine and what should it replace? *Nature Reviews Gastroenterology & Hepatology*, 2012.

[6] Margaret A. Hamburg. The Path to Personalized Medicine. *The NEW ENGLAND JOURNAL of MEDICINE*, 2010.

[7] Richard L. Schilsky. Personalized medicine in oncology: the future is now. *Nature Reviews Drug Discovery*, 2010.

[8] Jerel C. Davis et al. The microeconomics of personalized medicine: today's challenge and tomorrow's promise. *Nature Reviews Drug Discovery*, 2009.

[9] Greifswald University Hospital, Institute for Community Medicine. Projekt GANI_MED (Greifswald Approach to Individualized Medicine) (Last accessed: Dec 19, 2013). `http://www.medizin.uni-greifswald.de/GANI_MED/index.php?id=606&L=1`.

[10] Institute for Community Medicine (Last accessed: Jan 23, 2014). `http://www.medizin.uni-greifswald.de/icm/index.php?id=19&L=1`.

[11] Völzke E. Study of Health in Pomerania (SHIP). Concept, design and selected results. *Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz*, 2012.

[12] Martin Langanke. Comparing different scientific approaches to personalized medicine: research ethics and privacy protection. *Per Med.*, 2011.

[13] Andreas Bauer. *Data Warehouse Systeme - Architektur, Entwicklung, Anwendung*, volume 4. 2013.

[14] Erhard Rahm and Hong Hai Do. Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin*, 23:2000, 2000.

[15] National Information Standards Organization. Understanding Metadata. *NISO Press*, 2001.

[16] Dortje Löper. Das Entity-Attribute-Value-Konzept als Speicherstruktur für die Informationsintegration in der ambulanten Pflege. *INFORMATIK 2011 - Informatik schafft Communities, 41. Jahrestagung der Gesellschaft für Informatik*, 2011.

[17] Oracle. Logical Database Limits (Last accessed: Dec 26, 2013). `http://docs.oracle.com/cd/B19306_01/server.102/b14237/limits003.htm`.

[18] Cynthia A. Brandt et al. Metadata-driven creation of data marts from an EAV-modeled clinical research database. *International Journal of Medical Informatics*, 2002.

[19] Vojtech Huser. GANI_MED Strukturbereich 3: Medizininformatik und IT-Kohortenmanagement.

[20] Heidelberg University Hospital. Relationship ETL process and data warehouses, internal document, 2013.

[21] Greifswald University Hospital, Institute for Community Medicine. Research database values structure, internal document, 2013.

[22] Greifswald University Hospital, Institute for Community Medicine. Web application at transfer unit (developmental version), internal document, 2013.

[23] Greifswald University Hospital, Institute for Community Medicine. BPMN process of data application, internal document, 2013.

[24] Greifswald University Hospital, Institute for Community Medicine. Process of data application, internal document, 2013.

[25] Greifswald University Hospital, Institute for Community Medicine. G2D2 DTO model, internal document, 2013.

[26] The Eclipse Foundation. Eclipse IDE (Last accessed: Dec 19, 2013). http://www.eclipse.org/.

[27] Appcelerator. Aptana Studio 3 (Last accessed: Dec 19, 2013). http://www.aptana.com/products/studio3.

[28] Oracle Corporation. What is Java technology and why do I need it? (Last accessed: Jan 8, 2014). http://www.java.com/en/download/faq/whatis_java.xml.

[29] Wikipedia. Java (programming language) (Last accessed: Jan 8, 2014). http://en.wikipedia.org/w/index.php?title=Java_%28programming_language%29&oldid=589657800.

[30] Wikipedia. JavaScript (Last accessed: Dec 19, 2013). http://en.wikipedia.org/w/index.php?title=JavaScript&oldid=586706332.

[31] Oracle. Oracle Database Express Edition (Last accessed: Dec 19, 2013). http://docs.oracle.com/cd/B25329_01/doc/install.102/b25143/toc.htm.

[32] Canonical Ltd. Ubuntu (Last accessed: Dec 19, 2013). http://www.ubuntu.com.

[33] Wikipedia. UML (Last accessed: Dec 19, 2013). http://en.wikipedia.org/w/index.php?title=Unified_Modeling_Language&oldid=586682636.

[34] Partners Healthcare. Informatics for Integrating Biology and the Bedside (Last accessed: Dec 19, 2013). http://www.i2b2.org.

[35] Partners Healthcare. Informatics for Integrating Biology and the Bedside (Last accessed: Dec 19, 2013). https://www.i2b2.org/software/files/PDF/current/HiveIntroduction.pdf.

[36] Vikrant G Deshmukh, Stéphane M Meystre, and Joyce A Mitchell. Evaluating the informatics for integrating biology and the bedside system for clinical research. *BMC Medical Research Methodology*, 2009.

[37] Partners Healthcare. Release Notes for i2b2 Version 1.6.xx (Last accessed: Dec 19, 2013). `https://www.i2b2.org/software/releaseNotes_current.pdf`.

[38] Thomas Ganslandt, Ulrich Sax, Matthias Löbe, Johannes Drepper, C. Bauer, B. Baum, J. Christoph, S. Mate, M. Quade, Sebastian Stäubert, and Hans-Ulrich Prokosch. Integrated Data Repository Toolkit: Werkzeuge zur Nachnutzung medizinischer Daten für die Forschung. In Ursula Goltz, Marcus A. Magnor, Hans-Jürgen Appelrath, Herbert K. Matthies, Wolf-Tilo Balke, and Lars C. Wolf, editors, *GI-Jahrestagung*, volume 208 of *LNI*, pages 1252–1259. GI, 2012.

[39] Sebastian Mate. i2b2 Wizard (Last accessed: Dec 19, 2013). `http://www.imi.med.uni-erlangen.de/tools/i2b2-wizard/`.

[40] Partners Healthcare. i2b2 Clinical Research Chart (CRC) Design Document (Last accessed: Dec 19, 2013). `https://www.i2b2.org/software/projects/datarepo/CRC_Design_Doc_13.pdf`.

[41] Eli Grey. FileSaver.js (Last accessed: Dec 19, 2013). `https://github.com/eligrey/FileSaver.js/`.

[42] Katharine Miller. NCBCs Take Stock and Look Forward: Fruitful Centers Face Sunset. *Biomedical Computation Review*, 2012.

[43] TMF e.V. A Toolkit for Using the i2b2 Platform (Last accessed: Dec 19, 2013). `http://www.tmf-ev.de/EnglishSite/News/articleType/ArticleView/articleId/1277.aspx`.

[44] Segagni D, Tibollo V, Dagliati A, Zambelli A, Priori SG, Bellazzi R. An ICT infrastructure to integrate clinical and molecular data in oncology research. *BMC Bioinformatics*, 2012.

[45] Mate SN, Bauer C, Baum B, Engel I, Löbe M, Prokosch HU, Quade M, Sax U, Stäubert S, Winter A, Ganslandt T. The Integrated Data Repository Toolkit (IDRT). *i2b2 Academic Users' Group*, 2013.

[46] Majeed RW, Röhrig R. Automated realtime data import for the i2b2 clinical data warehouse: introducing the HL7 ETL cell. *Studies in health technology and informatics*, 2012.

[47] Majeed RW. HIStream (Last accessed: Jan 9, 2014). `http://sourceforge.net/projects/histream/`.

[48] Partners Healthcare. i2b2 Webclient (Last accessed: Dec 19, 2013). `https://community.i2b2.org/wiki/display/SMArt/SMART-i2b2+web+client`.

[49] Greifswald University Hospital, Institute for Community Medicine. ETL-process for the research database, internal document, 2013.