Technology
Arts Sciences
TH Köln

firely

Master Thesis

Evaluation of Methods for
Querying, Filtering and Aggregating
Complex Health Care Data
with regard to their Applicability in FHIR

Alexander Zautke, B.Sc. - 11096037

Faculty:	F10 - Informatik und Ingenieurwissenschaften
Field of study:	Master Computer Science - Software Engineering
Supervisor:	Prof. Dr. Heide Faeskorn-Woyke
External examiner:	Ardon Toonstra, M.Sc.
Company:	Firely Bos en Lommerplein 280 1055 RW Amsterdam
Date of submission:	16.08.2018

Declaration of Originality

I declare herewith, that I am the sole author of this thesis.

Furthermore, I confirm that:

- this work has been composed by me without assistance;
- I have clearly referenced in accordance with departmental requirements, in both the text and the bibliography or references, all sources (either from a printed source, internet or any other source) used in the work;
- all data and findings in the work have not been falsified or embellished;
- this work has not been previously, or concurrently, used either for other courses or within other exam processes as an exam work;
- this work has not been published.

I confirm that I understand that my work may be electronically checked for plagiarism by the use of plagiarism detection software and stored on a third party's server for eventual future comparison.

Gummersbach, 16.08.2018

Abstract

Objectives FHIR (Fast Healthcare Interoperability Resources) is an emerging standard for improving interoperability in the domain of health care. Besides offering features for achieving syntactical, semantical and organizational interoperability, it also specifies a RESTful API for searching purposes. The main objective of the following thesis revolves around investigating open challenges and limitations of the so-called FHIR Search Framework.
Methods A variety of operations for searching in FHIR resources, including all search interactions, definitions of search parameters, search parameter types and advanced search concepts are described. Over the course of the thesis, a quality model based on ISO 25010 is established. It serves as the foundation for determining if the FHIR Search Framework is well-suited to cover the information needs of its users. An analysis of completeness involving the measures defined in the quality model forms the main contribution. The primary discussion of the research questions is concluded by proposing a graph model for determining reachability between FHIR resources, essentially mirroring the chaining and reverse chaining functionality. Using well-known classes for expressiveness in graphs, the thesis assess to which degree a graph search can be formulated with the currently defined capabilities.
Results From a functional perspective the FHIR Search Framework can be considered well-suited. Practical limitations should be minimal, grounded on the fact that extensive coverage of the lowest expressiveness classes, RPQs and 2RPQs, can be achieved. Severe gaps where identified only in the support of C(2)RPQs and Data Path Queries. Additionally, ideas for improving non-functional aspects are introduced to support developers in learning the standard and testing their implementations.
Conclusion The evaluation of the FHIR Search Framework showed promising results in terms of functional completeness. Yet, the standard is still evolving, and certain parts of the Search API are neither well-known nor implemented widely. A discussion is to be held if the specification should cover more sophisticated aspects that result from the gaps which were identified.

Keywords:
Interoperability; FHIR; Health Care; Quality Models; Graph Query Expressiveness

Contents

1. Introduction

1.1. Motivation

Fast Healthcare Interoperability Resources (FHIR) is an emerging standard, published and created by HL7 [1], that is set out to improve interoperability in the domain of health care. To facilitate this goal, the FHIR specification provides definitions of so-called “resources” as one of its main components. Resources are representations of concrete entities and concepts[1], which are commonly involved in processes of the setting mentioned above. Through the use of such a data format and its elements, complex health care data can be modeled in a consistent, modular and extensible way. In combination with a set of clearly defined APIs, it is possible to interoperably describe, exchange and store relevant information. [2, p.900]

As a result, such an approach could lead to improvements regarding the interaction with health care data from the perspectives of various stakeholders [3]. To introduce the concept and benefits of interoperability, a selection of use cases, involving an exchange of health care data, is outlined:

- **Patients & Clinicians**
  Through interoperability, patients and clinicians are enabled to effectively and efficiently retrieve data from a diverse set of health care providers and connected institutions, such as laboratories. Thus, instead of having to rely on incomplete data, i.e., data that is only locally and directly available, a complete and comprehensive overview of the patients’ health care situation can be gained. [4]
- **Researchers**
  In research, health care data is increasingly re-purposed for other processes than direct patient care. For example, medical decision support is fostered by the provision of high-quality medical evidence, which is gathered from data sources consisting of de-identified clinical data. In these cases of secondary use, it is crucial to include interoperability as part of an overall implementation strategy in order to maximize the advantages across different institutions and research fields. [5]
- **Developers & End-users**
  By developing and deploying an unified interface to health care systems, it is possible to provide benefits for developers and end-users as well. Mainly, by being able to access a standardized API and use a harmonized set of related frameworks and tools, it can be ensured that a common development platform is established. Therefore, instead of being forced to write applications that are specific to a health care environment, implementations can be shared across institutions and executed in different contexts. Furthermore, end-users are no longer restricted to a specialized set of applications, which were aligned to their system, but can freely choose from a range of applications to cover their needs. To encourage the development of interoperable solutions and to foster the distribution of these applications, App Store like networks can be established [6]

[1] For example: “Patients” or “Observations”.

Summarizing the promised benefits, it can be stated that, if successfully alleviate deficiencies resulting from a currently less well-developed interoperability, improvements in numerous areas of health care (including clinical operations, public health, and evidence-based medicine) could be achieved. These advances result from the fact that, due to interoperability, the respective data can be analyzed, combined and compared with minimal effort, but most importantly be understood correctly by each recipient.

In the recent past, FHIR has been successfully implemented in a variety of services and products [7]. Moreover, it has become the focus and subsequently the foundation of national health care interoperability strategies. Examples for this development include the Argonaut Project [8] in the USA, which is based on the recommendations presented in [9], or the MedMij project in the Netherlands [10]. Due to the growing use of FHIR, it is now an ongoing task to integrate this next-generation standard into existing health care landscapes. Besides the fact, that the integration of standards was seen as a challenge in the past [2, p.899], initial implementations of FHIR have shown promising results [11], [12].

However, additional new challenges besides interoperability have arisen in the context of today's electronic health record systems (EHRs), namely how to handle big data requirements. Most notably, data sources now potentially include 3D-Imaging data, various (real-time) sensor readings and even genomic information. This diversity results in the opportunity to provide personalized health care by having an improved foundation for making medical decisions. To effectively and efficiently utilize this data in a broad context, strategies need to be defined, describing how to deal with the volume, velocity, and variety of the aforementioned health care data [13].

By focusing on interoperability, FHIR inherently facilitates the handling of health care data with regard to their variety. Additionally, the FHIR standard supports the option to search resources through a framework in the form of a REST API [14]. Consequentially, it is possible to manage the complexity resulting from the increased information load, by detailedly specifying which data is needed. Therefore, clients using the FHIR Search Framework may potentially be equipped to comprehensively tackle demands, as the option to define relevant selection criteria is given.

Besides having this option, it needs to be taken into consideration that a number of different ways to search, filter, query and aggregate data sources have been proposed in literature. Amongst other, the methods discussed below have been developed in general. The current thesis aims at contributing towards a discussion about the limitations of the FHIR Search Framework, as well as, how the hereinafter described methods can provide coverage of additional use cases around the analysis of health care data.

- **MapReduce Views**
  A MapReduce view provides a virtual perspective on a specific dataset that is to be queried, meaning that the key focus of this method lays on defining a way of transforming input data into a desired output structure that can be looked upon request. As a result of this, MapReduce Views provide the possibility to sort, filter, and aggregate said data through user-defined functions. Examples of MapReduce View implementations can be found in [15], [16].

- **Query Languages for semi-structured data**
  Query languages, like exemplarily defined in [17], [18], provide the opportunity to query semi-structured data (i.e., data that does not conform to a formally specified structure but nevertheless contains essential semantical elements) in a declarative way. Therefore, this method enables specific parts of arbitrary data formats, like XML or JSON, to be located by traversing the data. This selection can then be filtered and transformed.
- **Query Languages for specific data models**
  This category refers to query languages that were designed to be used in combination with specific types of data models, like Graph or Relational models, in order to incorporate the underlying properties of the models into the languages' design. In contrast to this, query languages exist, that also provide the possibility to be used in multi-model contexts, i.e., databases that support multiple data models in parallel. An initial selection of data model specific query languages is discussed in [19, p.50ff.].

Due to the fact that other methods for searching in data and related operations exist, which may support these tasks more broadly and conveniently, the primary goals of this thesis emerge. In order to provide a sound basis for evaluating the appropriateness of the currently chosen REST API for retrieving resources, the following approach is adopted:

1. The thesis seeks to provide clarity about the aspects in which the current FHIR Search Framework exhibits open challenges in its effectiveness and efficiency.
2. If improvement potential exists within the FHIR Search Framework, it aims at providing a structured analysis of alternative methods and discusses how to close potential gaps by combing these methods with current FHIR systems.
3. Resultantly, this thesis aims at contributing towards a comprehensive guiding framework that assists in choosing an appropriate method for seeking information in FHIR-supporting systems while retaining overall interoperability.

Please note that it is not intended to deliver a final recommendation for an approach how to handle the challenges resulting from searching and similar interactions in the FHIR specification. Instead, the thesis explores alternatives and provides constructive critique regarding the individual methods to provide a basis for making such a decision.

1.2. Research Questions

After having outlined the intentions of the current thesis, a substantial and objective factual basis regarding the involved topics is to be achieved. First and foremost, this aim is to be accomplished by answering the following research questions:

1. Is the current FHIR Search Framework well-suited to solve tasks and challenges occurring within its current scope?
2. How can stakeholders be empowered to cover their needs and wants for searching, querying, filtering and aggregating complex health care data within in the context of FHIR?
3. Can any additional method for searching, querying, aggregation and filtering be used to bridge substantial gapes occurring within the FHIR Search Framework?

A discussion of the exact motives behind the selection of these research questions can be found in the next sections.

1.3. Scope & Limitations

To provide a legit basis for the validity and accuracy of the results presented within this thesis, it is acknowledged that the research is based on the assumption that open challenges exist within the FHIR Search Framework and that the introduction of other approaches to search-related operations could substantially enhance the standard. However, the possibility that the current REST API approach is broadly sufficient should be evaluated transparently.

The following section provides an overview of the strategy for the conducted research which should be perused to ensure a solid approach. A visualization of the research strategy can be found in Figure 1.

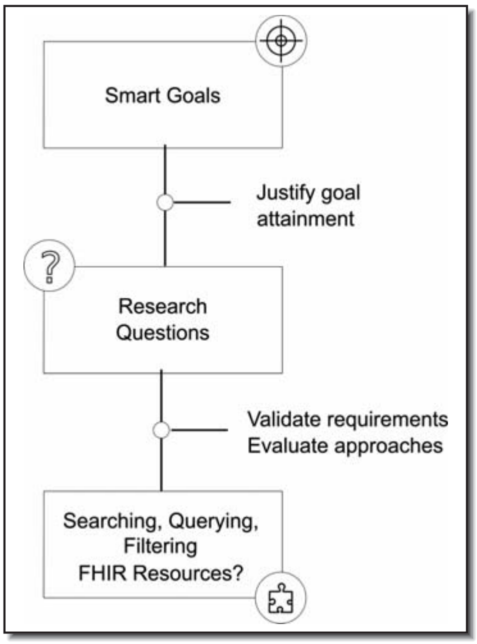


Figure 1: Research Strategy (Goals & Methods)

In general, the research questions take, as depicted, a central role. Their intention is twofold. First, the questions aim considerably at collecting practically applicable background information about the current FHIR implementation landscape, such as:

- Is the focus of the FHIR Search Framework aligned with the demands and expectations of implementers?
- Is the FHIR Search Framework well-understood and in use?

Through a resulting analysis, functional requirements that exist in practice should be determined, specifying how needs should be covered to provide a sufficient degree of effectiveness with regard to searching in health care data. Moreover, it should provide a starting point for researching any areas of future improvements and highlight currently unsupported demands.

On the other hand, the research questions are set out to provide a foundation for the theoretical evaluation. A set of generally applicable features in the context of searching should be identified, to investigate an extension of the FHIR Search Framework, such that it can potentially support existing use cases in a more elaborate way. Additionally, it is to be investigated based on which objective measures, shortcomings in the design of the FHIR Search Framework can be determined.

A structured summary of supporting methods is depicted in Figure 2, showing through which steps, answers to the research questions are being gathered. In general, by following this research structure, it is expected to reach a transparent basis for quantifying certain terms specified in the research questions. Hereby, it is in particular answered how effectiveness, efficiency and appropriateness - key to central issues within this thesis - is to be interpreted in the context of the FHIR Search Framework. As a result, the overall degree of the thesis' goal attainment should be determinable, as the research question are designed to concretise the research goals.

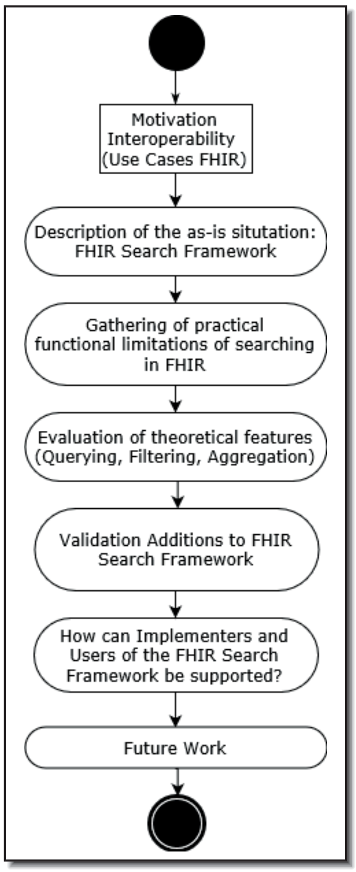


Figure 2: Evaluation of Methods for Querying, Filtering and Aggregating in FHIR

It should be noted that for the above-mentioned evaluation, respectively any resulting recommendations, specific constraints exist, namely that the fundamental design principles of FHIR [20] should be satisfied. These self-given priorities and guidelines include the following:

- **FHIR prioritizes implementation**
  The key audience of the FHIR specification are implementers. This focus becomes particularly evident with regard to the specifications' design decisions. Instead of striving for complete coverage of its clinical background or providing an "ideal" standard architecture, it aims to simplify the overall implementability to all intents and purposes. Details about how this principle is being applied when designing the FHIR specification and its API can be found in section 2.2. Thus, any changes or newly added (query, search, or filter) methods should follow this principle as well, such that it can support implementers to overcome their challenges and not hinder them through adding complexity to their systems. Additionally, instead of providing only a theory-focused evaluation in this thesis methods, concrete and implementable recommendations should result.
- **FHIR keeps complexity where it belongs**
  By applying an "80/20" principle, the FHIR specification tries to support features used by roughly 80% of the implementers, whereas the other 20% of functionality can be added via custom extensions. The same design rationale should be applied when evaluating any methods within the scope of this thesis. It helps to achieve a balance between theoretical features and their practical applicability.
- **FHIR leverages common (web) technologies**
  Instead of inventing custom solutions, the FHIR specification leverages existing methods and techniques where possible. Therefore, instead of, e.g., focusing on inventing a specialized FHIR query language, existing methods, design patterns and best practices should be evaluated first to determine to what degree they can sufficiently support the FHIR Search Framework. However, a custom solution is not to be dismissed entirely.

1.4. Structure of the Thesis

For improved navigability, the thesis' structure and its general line of argumentation is outlined as follows:

The current documentation is structured into seven separate parts, which mainly correspond to the presented research process of the previous section. The first chapter starts off by portraying the motivation for the carried out research. Especially, use cases where interoperability is needed as part of the foundation of health care systems are discussed briefly to introduce often occurring challenges. Additionally, the current need of various stakeholders to search in health care data is exemplified. Subsequently, research questions, as well as methods for answering them in a structured and transparent way, are derived from the problem description of this thesis and hereupon based goals.

The second chapter provides in-depth theoretical background information about the main topics, which are needed to extensively understand the context of the presented findings of this thesis.

Chapter number three provides a summary of the as-is situation by describing the FHIR Search Framework and its current scope, features, and design principles. Chapter four continues with the discussion of this approach to searching in FHIR. It asses the completeness of the FHIR Search Framework by categorizing its functionality using a “quality in use” model. For each group of related elements, measures are postulated which can be used to determine practical limitations of the specified capabilities. This requirements engineering phase is completed by analyzing which non-functional factors are influencing the usability of the FHIR Search Framework.

Chapter five focuses on practically improving the general applicability of the FHIR Search Framework. It focuses on usability measures by which the necessity of fundamental changes are to be evaluated.

Chapter six discusses how to help implementers and users of the FHIR Search Framework to overcome current challenges and solve advanced use cases effectively and efficiently. It takes other search, query and filter methods into consideration that are currently not part of the FHIR search framework. Based hereon, it discusses how these methods could enhance the FHIR standard in cases where the current approaches fail.

Chapter seven concludes this thesis by providing an outlook on which topics are to be researched more in-depth. Additionally, it summarizes the core aspects of the FHIR Search Framework and where it should focus on, such that all involved stakeholders can reliably and appropriately deal with their medical data.

1.5. About the Company

To provide a better understanding of the thesis’ goals and for transparency reasons, a brief description of the organizational environment of this thesis is given.

Firely is an Amsterdam based company, which has been involved with FHIR since 2012. It consists of software engineers, support engineers, and consultants, who are participating in the development of tools, servers, and platforms related to FHIR. Additionally, Firely is actively contributing to the FHIR standard itself, through participating in HL7 Working Group Meetings and having one member of the FHIR core team as part of the company. Moreover, Firely provides services to vendors, care providers, and governments who are seeking to adopt FHIR for their use cases.

By developing and providing the abovementioned products and services, Firely is interested in seeking feedback regarding challenges and open issues related to FHIR. Consequently, the company is interested in the outcomes of the current research and therefore provides support for this thesis.

2. Theoretical Background

Despite the challenges in health care resulting from interoperability concerns and Big Data, it is important not to neglect the reason why effective and efficient communication between the various involved stakeholders is to be achieved in the first place: health care should be assisted in delivering optimal patient care. In this context, health care IT and especially interoperability are a cornerstone in achieving high-quality health care, as needed computational and organizational resources are provided through these means. Simultaneously, health care is improved in the following areas by introducing interoperability [21], [22]:

- Clinical Safety, by having a complete and uniform picture of the patient's health care situation.
- Patient satisfaction, resulting from quality improvements being achieved as research and therapy is combined more closely. In this context, pursuing the vision of achieving a learning health care system can be helpful, as such a health care system would allow all stakeholders to collaborate jointly in closed feedback loops to produce the best results possible for all patients and involved providers.
- Reliability, by using standardized solutions that were validated and officially balloted by experts instead of "home-brewed" structures.

Conversely, in cases of missing interoperability, situations in clinical contexts may occur that can be classified as wasteful, harmful or risky. These circumstances are most likely a factor leading to higher health care costs [23]. However, even in the light of these additional expenses, it is counterproductive to only see the patient as a cost factor that needs to be optimized. Instead, it is proposed in literature that health care should focus on achieving a "value" for the patient. Value is to be interpreted "[...] as maximum health benefit at minimum cost [...]" [24]. To practically realize this intention, including the benefits listed above, health care IT platforms are needed, which can be used as a supporting commodity [25].

To be able to fully embrace value-based health care and to foster the accompanying introduction of interoperability [26], support from an organizational point of view is needed as well. Instead of having a function-oriented focus, health care organizations have been prompted to engage in patient-centric care [27], [28]. This approach entails a focus on the patient's needs, instead of seeing the patient as an abstract part of the general clinical workflow. Moreover, the patient is encouraged to actively participate in the clinical processes, e.g., by sharing personally collected health data [29].

By empowering the patient to contribute to her / his own health and focusing on the patient's need, the health care system's socio-technicality is emphasized. More practical implications resulting from the combination of organizational structures and technology in health care can be found in [30]. Resultantly, it should be apparent that it is not satisfactory to only provide health care institutions with interoperable IT solutions and standards. Preferably, an interoperability strategy for a health care organization should acknowledge that organizational and technological aspects are impacted at the same time.

Bridging between the realm of the patient and the related but not necessarily supporting technology side can be enabled through interoperability. An overview of possible layers of interoperability is given in the following section. It starts off by discussing the drivers behind interoperability. To provide an shared understanding within this thesis, the ISO/IEC definition of interoperability is examined. After having established solid knowledge about these topics, section 2.2 outlines how interoperability is achieved in the FHIR standard. An introduction to database related issues and concepts, which are relevant in the context of this thesis, is concludingly provided in section 2.3 and 2.4.

2.1. Interoperability

In the context of health care, it is essential that there is a sharp focus on profoundly and harmoniously integrating IT systems into the overall organizational architecture, such that these systems can support a broad spectrum of use cases. Without aiming for this target, barriers for achieving synergy effects are indirectly established. Primarily, it results in the challenge that involved stakeholders tend to have negative attitudes regarding their systems, but have to cope with any deficiencies due to required clinical workflows [31]. This issue should illustrate the need for appropriate systems, meaning systems that support users effectively, efficiently and to a general satisfaction. These capabilities can be summarized, in accordance with the ISO/IEC 25010 standard [32], by the term “Quality in Use”. It attests the system’s ability to serve the users with its intended functionality. In general, the quality in use is fostered by an accompanying product quality, which defines measures for determining the internal and external qualities of a software system. Therefore, the software’s effects on its intended environment are determined by the former quality model, whereas the latter one specifies inherent characteristics.

With regard to these quality models and the nature of health care systems, a strong need for security, reliability, and interoperability can be identified, as patients data need to be secured, systems need to be continuously available, and stakeholders need to exchange and use data from various sources. This section focuses on the effects that result from the ISO/EC 25010 standard definition of interoperability. It is to be noted that the absence of any further elements of the quality models may be detrimental to health care systems. However, they are not within the scope of this thesis. More detailed information about additional quality characteristics can exemplarily be obtained in [33], [34].

With regard to one of the overall themes of this thesis - standardization - the internationally accepted ISO/IEC 25010 definition of interoperability is selected to establish a comprehensive understanding of this concept:

> “**Interoperability**. Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.”

This definition results in several implications:

- Interoperability is determinable as a degree. Therefore it is not a concept which is explicitly and solely a feature of a system, product or component, only if specific criteria are met. Instead, the real extent can be measured and evaluated.

- Interoperability needs to take different levels of communication into account. The requirement to use the exchanged information results in the obligation to not only achieve interoperability on a technical level but an organizational level as well, such that the information exchange becomes meaningful.
- Interoperability is not concerned with unifying the environment of the communication or the information systems itself. It is not inherently necessary that all connected systems, products or components in the network are homogeneous. Rather, the ability to achieve communication and understand the transferred information is gained by agreeing on a "common denominator". For example, in the context of health care, standards like FHIR take this role.

Due to the different connotations of interoperability, more extensive definitions have been developed in literature to highlight these associated aspects. A comparison of the various definitions can be found in [35].

In general, as stated above, interoperability can be divided into several levels. For the purpose of clarification, these levels and a description of their corresponding meaning and objectives can be found below. The extent of interoperability increases with each level, starting with technical interoperability as a minimum. By combining these levels as building blocks, an information exchange maturity schema can be established. Resultingly, in cases of high-level interoperability, a supporting interoperable foundation can be found as well [36]. By having this logical hierarchy, the reason for defining interoperability as a degree and not as a structural characteristic is justified.

1. **Technical / Foundational interoperability**
   Technical interoperability, which is also synonymously referred to as foundational interoperability, is for the most part concerned with linking all systems, devices, and components that are needed for an aspired exchange of information. Based on fitting communication protocols[2] and a conformant infrastructure, a machine-to-machine communication can be implemented. [37]
2. **Syntactical interoperability**
   Syntactical interoperability refers to the ability to verify the structure of exchanged information. By unambiguously defining a description for this purpose in the form of a logical model, it is possible for any communication party to determine if certain aspects are encoded correctly. In general, a standard, to which the content of an information exchange confirms to, should be used. It helps to determine applicable elements, their data types, cardinalities and basic validation rules. [38]
3. **Semantical interoperability**
   On the basis of the previous levels of interoperability, semantical interoperability engages in exhaustively and critically structuring the context of communication in a linguistic sense. In this way, concepts[3] that can be part or subject of the communication are uniquely labeled with identifiers. As a result, it is possible for all involved senders and receivers to understand the content of the communication in the same way, as it can directly be retrieved in a systematic way. [39, pp. 21ff.]

[2] Transport and Application protocols need to be selected to specify to whom and how information is send.
[3] Representational units that categorize all "things" that can be found in a specific domain.

4. **Organizational interoperability**
   Organizational interoperability is the ability of organizations to overcome technical and structural barriers and effectively share and use information across different information systems, work processes, organizational units, and even cultures. [40]

Concludingly, it can be stated that interoperability, especially in contrast to other quality characteristics like compatibility, mandates a tight integration and collaboration on the basis of recognized standards. In cases where such an approach is disregarded, it cannot be ensured that emerging technical and use-related challenges can be solved effectively and efficiently on a large scale, i.e., on a cross-organizational or even international level.

2.2. FHIR

The following section deals intensively with conceptual details about FHIR. Without discussing any specifics, it can be noted that this standard mainly consists of resources definitions and an appendant API for managing these representations of health care concepts. To understand the scope of FHIR and any resulting effects on implementers in detail, this section primarily focuses on providing an introduction to the main architectural elements of resources, the defined API interactions, applied technical paradigms and core principles. It aims at presenting how interoperability can be achieved on multiple levels by using this standard. More specifically, it is analyzed in the next sections through which methods health care information can be exchanged, what elements are available in resources to structure recorded data, and how these elements can be extended to contain computer-processable codes about the elements' meaning in health care. All presented details refer to FHIR release 3 (Standard for Trial Use), which is published as the official version at the time of writing of this thesis.

2.2.1. History of HL7 Standards

Due to a limited scope of this thesis, co-existing health care standards are not discussed in detail. This is particularly notable as FHIR is not the first well received or most expressive health care standard that is published, nor is it currently widely implemented within clinical contexts. However, industrial and community parties have selected FHIR as the candidate for the next primary standard of HL7 [41]. This emphasis of a standardization organization can be conceived as an indication for an area of further scientific research. In such a context it becomes objectively possible to validate the outreach, promises, and outstanding challenges of the current specification. These motives are amplified by the fact that open problems could still be addressed through such research while the standard has not completely reached a normative status. Nevertheless, it should be clearly stated that the presented features concerning interoperability are not a unique feature of FHIR or other HL7 standards for that matter. Alternative standards like openEHR [42] are respectfully acknowledged but are not subject to the research of this thesis.

Despite the imposed limitation in the scope of this thesis, a brief overview of the history of HL7 health care standards related to FHIR is presented. This abbreviated review should help the reader to have a better understanding of all considerations that were taken into account for the design decisions of the standard's current revision.

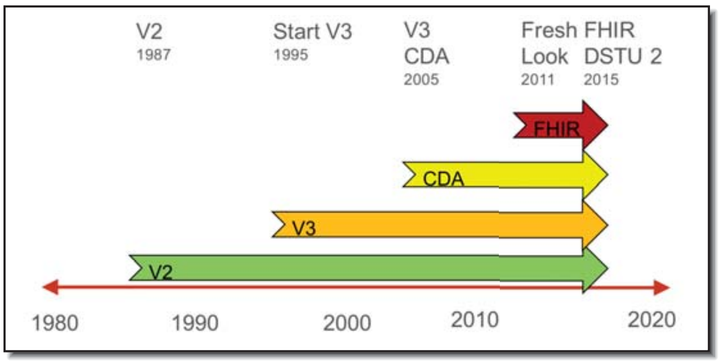


Figure 3: Timeline of HL7 Standards [43, p.17]
Licensed by CC BY 3.0

As depicted in Figure 3, HL7 was founded as a non-profit standards organization in 1987. Since then it pursued their mission of enabling secure access to health care information around the world, by publishing standards in this domain.

Having recognized the initial need for a common interface between hospital systems, HL7 published its first version of the widely adopted HL7 v2 standard in 1989, after delivering a proof-of-concept in 1987 [44]. Essentially, this standard enables the exchange of clinical messages based on trigger events, such as the admission, discharge or transfer of a patient.

The focus of HL7 v2 lays on specifying the content of messages in the form of so-called segments and the corresponding messaging syntax. These segments are ordered sets of data elements with associated data types, which are applicable to capture clinical information. Each message has a concrete message type and is encoded according to an abstract message syntax table, containing the sequence and cardinalities of its enclosed segments per message type. Each segment is represented as one line in the message. Per segment individual elements are placed at a predetermined position, which is deducible from the name of the element. For example, a PatientName (Segement PID-5) within a Patient Identification Details segment is placed at the 5th position of its segment. The position is measured based on special delimiters, separating the individual elements. The reader is referred to [39, pp. 223ff.] for more internal details.

Due to significant adoption of HL7 v2, it has generally been possible to detailedly evaluate the standard in production systems. Thereupon, a few significant shortcomings were identified in literature, which have been partially attributed to the fact that the standard has been developed in large parts by clinical specialists based on a more or less ad-hoc methodology. As presented in [45] and [46], the most notable drawbacks in the standard are related to a missing comprehensive, concise, and robust information model that backs the messages' content structurally and semantically. Consequently, HL7 v2 does not exhaustively cover the representation of concepts in a wide-ranging selection of clinical subdomains[4].

[4] For example: Medications, Financials, Clinical Workflows.

These deficiencies can practically be validated by comparing which FHIR elements can be mapped to a matching v2 concept. Each FHIR resource specifies these mappings if available. It is to be noticed that a significant portion of resource elements cannot be mapped directly. As a workaround, the less-than-ideal solution of extending the content of the messages proprietorially can be used. This extension is to be realized through freely definable Z-segments. When exchanging such a message segment, take care must be taken to achieve semantical interoperability by communicating the meaning of the segment itself and the codes used within it.

As a successor to v2 messaging, HL7 v3 and the Clinical Document Architecture (CDA) were developed, which are based on a fundamentally different concept: both standards instate a model-driven design approach. In abstract terms, HL7 developed three major components which in combination allow deriving a strict and transparent specification for the exchange of clinical information [47]:

- **Reference Information Model (RIM)**
  The intention of the RIM is to provide a complete information model which abstractly represents, through the use of UML, in which contexts persons and things interact in the domain of health care. This interrelation is expressed in two parts:
  - By modeling roles, participation, and acts that can be adopted and executed by the entities in occurring encounters
  - By specifying the relations between roles, acts and their respective links

  As a result, the RIM provides base classes, as well as concrete specializations (e.g., patient as a specialization of a role), which offer attributes for all of the previously mentioned cases where information need to be recorded.
- **HL7 v3 Datatype Specification**
  The Datatype Specification is introduced in addition to the RIM to define the structure and constraints of the class attributes in the information model. Through these formal definitions of available types, attributes in the specification become unambiguous regarding their semantics and can be computationally interpreted.
- **Vocabulary Specification**
  Specific attributes in the RIM can only be populated from a predefined collection of codes. In these cases, the Vocabulary Specification provides a set of accepted values in the form of internal unique identifier or references to external code systems (nomenclatures).

The mentioned artifacts allow information to be derived from the RIM and be exchanged using XML-based messages (HL7 v3 messages) or self-contained clinical documents (CDA documents).

Based on the capabilities provided by these specifications, the goal was set out to establish the RIM as the universal backbone for all health care information exchanges [39, p.244]. In hindsight, selecting a complete coverage of the health care domain as a strategic goal leads to the challenge of balancing between the needs of implementers and the detailedness of clinical modelers. Instead of concrete artifacts, an abstract health care model was developed, which must be constrained on multiple levels with custom tools to be implementable.

Out of a situation of conflicting requirements and deep conflicts about the suitability of the chosen approach on a technical level[5], HL7 set out the task of developing a “Fresh Look” initiative. The possibility of choosing a new approach to exchanging health care information without conceptional restrictions resulted in a preliminary specification (Resources for Health [50]), eventually providing the foundational concepts of FHIR in its current form [51].

2.2.2. Technical Interoperability in FHIR

Before analyzing in detail through which different methods health care information can be exchanged in FHIR, it needs to be stated that the standard in itself is not necessarily restricted to be only used in combination with a specific software architectural style. Instead, it is independent of such assumptions in every respect [43, p.44].

Instances of a standard-conformant API implementation can be deployed for example in exchanges between multiple back-end systems, when synchronizing information between EHRs, or be used as “brokers”, translating different health care standards or legacy database formats to FHIR. Moreover, it possible to use FHIR resources as a system internal model for health care data, without needing to open up the system with an external way of retrieving information. In addition to these use cases, FHIR could be used as a gateway for a vendor-neutral repository, where health care information may be modeled via FHIR resources and all requests from and to different systems are transformed to FHIR requests [43, p.46].

In all of the cases mentioned above, FHIR enables a foundational level of technical interoperability by supporting multiple paradigms for exchanging resources. Depending on unique requirements and existing system structures, multiple exchange methods can be chosen for this task. Their most representative characteristics include the following:

- **HTTP REST**
  One significant property of FHIR is its focus on being computable, meaning that parts of the API and the resources itself are aligned to minimize the effort for automatically achieving a rudimentary degree of interoperability. As a consequence, it is, for example, possible to structurally describe the capabilities of a specific FHIR-enabled API endpoint through a so-called CapabilityStatement resource [52]. Within such a description, systems operators can specify which resources can be handled by a system, which API interactions are provided in combination with these resources at a specific base URL, and how security requirements can be fulfilled by a requesting client. All in all, it is transparently documented how systems can interact, without needing further clarification, as a CapabilityStatement can be retrieved through the REST API (Metadata operation).

  To accomplish a broad degree of technical interoperability, the REST API provides the option to manage and enquire aspects of a system on different levels: distinct interactions are defined on an instance, type, and system level.

[5] For an intensive discussion please see [48], [49].

The instance level refers to the management of individual resources, including a variation of "CRUD-operations" (Read, Update, Patch, Delete). These interactions operate on a resource with a concrete id that was serialized in either XML or JSON. Moreover, it provides an option to retrieve the history of a particular resource. Especially in the light of the often occurring requirement in EHRs, to be able to inspect an audit trail [53], meaning to retrace what actions were performed on a specific set of information and what aspects have been altered, systems may choose to support explicit versioning. On a type level, resources can be created by a server and all resources of a specific resource type can be searched. An in-depth discussion of the search interaction can be found in chapter 3. System-level interactions contain amongst other, like a search and history operation across all resources, an interaction to process multiple API requests in a batch mode. Finally, these standard API interactions can be extended by a server, as custom operations can be defined, offering to execute actions on resources otherwise not defined by the standard.

  Examples of different REST API requests can be found in Appendix A. The full specification can be found at [54].

- **Messages**
  Regardless of the scope of the REST API, there exist situations where such an interface may not be applicable due to individual requirements. Examples could comprise an information exchange where the health care resources shall be exchanged efficiently but in an asynchronously way. Another example could encompass an exchange which shall be executed via a different medium than HTTP. In these circumstances, the messages paradigm of FHIR can be used.

  Analogous to HL7 v2, the FHIR standard defines how to transfer health care information through a loosely coupled request/response framework [55]. As detailedly explained in [56], there exist multiple categories of messages which can be sent, or, respectively, on which systems may choose to react. In general, messages are transmitted after clinical events, which are exhaustively and exclusively defined by the standard, have been triggered. Either a request for consequential actions (Consequence message), an information request based on queries (Currency message), or an indication that a specific activity was executed, whereupon a receiver may want to react (Notification message), can be dispatched from the event source to a known destination. A message consists of a MessageHeader resource indicating the type of the event, the source of the event, the target of the message and a description of the expected response. Additionally, it is defined what the main content of the message is in the form of focus resources, whose content is referenced in the header.

- **Documents**
  In certain situations in health care, clinical documents need to be created and distributed, for example, when giving patients access to their final discharge summary. These documents have distinctive characteristics, which need to be reasonably acknowledged when representing them digitally. These properties include the inability to alter the document, presenting the content in a form that enables it to be legally authenticated, and be human-readable [57].

To ensure the confidentiality, governance, and accuracy of the content of the document, FHIR provides a special resource and API to create immutable clinical documents based on a set of resources. At the center of a document is a Bundle resource which aggregates multiple individual resources. The order of these resources is provided by a Composition resource [58] which contains references to resources in separate sections. This Composition is located, like a "Table of Content" at the beginning of the bundle. Afterward, the full content of the referenced resources is included. The document is finalized by a narrative and signature.

- **Service Oriented Architecture (SOA)**
  SOA is not a concrete method for exchanging data. Instead, it describes an architectural paradigm for encapsulating services, which represent discrete functional units, executing defined operations over a network, independently of a particular business context. In health care environments it is not uncommon to find reoccurring activities of workflows, such as the registration of a patient or documenting problems and diagnoses [59]. In order to avoid redundancy when implementing these business processes, SOA can be used to decouple foundational functions, needed to perform these activities (e.g., create a patient record).

  Therefore, functions that are common to multiple activities can be bundled into separate services. Complex business use cases can then be supported, as higher level business needs can be efficiently solved by orchestrating SOA services, i.e., invoking and combining different services [60]. FHIR offers support for SOA by allowing to use FHIR resources to be passed between services [61]. The concrete exchange method and related communication protocols are however freely selectable, as SOA does not define any restriction in this regard [43, p.40].

2.2.3. Syntactical Interoperability in FHIR

FHIR provides a well-rounded set of resources as means to achieve syntactical interoperability within a wide range of fields related to health care. In consequence, verifiability of the exchanged information becomes feasible, as resources describe schemata which need to be adhered to for being FHIR-compliant.

Resources defined by the standard can be sorted into specific categories, all of which represent a distinct "theme". Most resources are concerned with one of the following objectives:

- Defining the structure and other elementary aspects of FHIR (Conformance and Infrastructure resources)
- Denoting a concept, which depicts a connection to a subject, object or element that has a physical manifestation within a health care system (Administration resources).
- Providing the ability to capture health care information for record-keeping or communication purposes (Clinical, Diagnostics, Medications, and Financial resources).
- Representing gained knowledge based on clinical information for sharing and evaluation purposes (Clinical Reasoning resources).

Resources are commonly derived from a general-purpose DomainResource [62], which generically specifies the structure seen in Figure 4. This assertion holds true for all resources, except for a few foundational resources, which differ due to technicalities. As a result, all formerly mentioned resources can contain the following content:

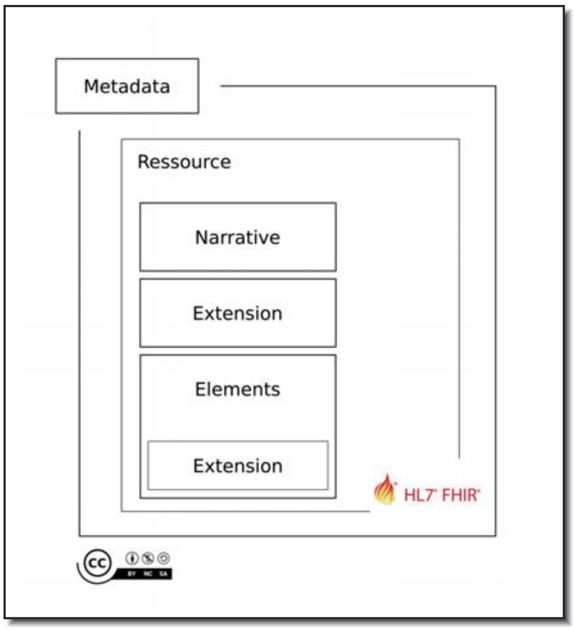


Figure 4: Components of a FHIR Resource [63]

- **Metadata**
  These fields provided can be used to provide contextual, non-content-oriented information, such as a literal identity assigned by the system for identification purposes in the form of an unique URI, or the date when the resource was last updated. Metadata facilitates the management of the resources without substantially affecting their content and interpretation [39, p.365].
- **Narrative**
  A human-readable summary of the content provided by the elements of the resource. Such a description can be used to provide a base level of interoperability as the textual representation of the resource elements can be interpreted by a human person as a last resort.
- **Elements and Extensions**
  As stated in the introduction of this thesis, FHIR provides diverse elements in a resource, which form the properties of an resource instance and capture the health care information. Either all necessary information can be described per use case by these elements directly, or they can be represented through added extensions.

  Additionally, it is noteworthy that the structure of resources is defined recursively. Resources are expressed in the standard through a StructureDefinition resource [64].

> Due to this approach, it is possible to computationally analyze a resource including its elements, their data types, any defined extensions, and constraints for implementation[6]. All in all, FHIR provides multiple so-called conformance resources which help to specify and validate the adoption of FHIR in a specific context, through mechanisms like extensions or so-called “profiles”. Detailed information about this topic is presented in the next sections, as conformance resources are more related to semantical and organizational interoperability than to the current topic.r

From a general point of view, it can be stated that resources are per se self-contained and meaningful. Based on this, when verifying a resource, is not necessary to further consult other resources, as all information are present within the resource itself and the profile it conforms to.

Nonetheless, it can still be valuable and necessary to interpret a resource within a specific context. An interpretation of a clinical observation can be regarded as a suitable example. Without knowing the circumstances leading to a particular measurement or assertion, it cannot be safely evaluated.

For the purpose of achieving an integrated overview, FHIR provides elements, where appropriate, which can represent “links” between resources. This mechanism can be adapted instead of bundling resources into a document, as described in previous sections. An exemplary use case for combining resources is shown in Figure 5.

In these cases, either a literal reference to another resource or a logical reference[7] identifying a concept which could potentially be exposed as a resource can be used [65]. The first option combines the FHIR resources through their literal identity. This identification property is gained, in the form of a Uniform Resource Identifier (URI), upon successful creation of the resource. Each resource is assigned a known identity and location in this manner. As it is an identification property, the latest version of the resource can “normally” be retrieved, as depicted in Figure 6, by combining the base URL of the used FHIR server, the resource type, and the mentioned potentially randomly chosen logical identifier. However, care should be taken when resolving literal references, as it is not a requirement that the server located at the base path of the URI is a (reachable) FHIR-API endpoint.

[6] Allowed element cardinality and logical flags indicate further information about an element, e.g., which elements must be supported by a server, which elements can be extended or are restricted for derivation, and which data types are allowed for a resource element.

[7] An arbitrary business identifier, like a National Patient ID, which resolves to one concrete entity. Please note that this is not limited to a Patient, but can be used for all kinds of concepts.

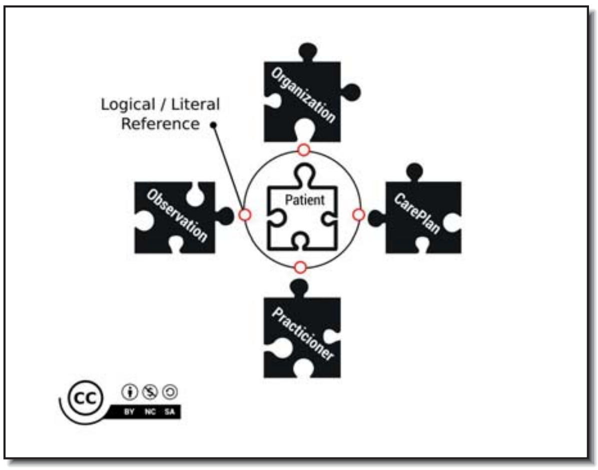


Figure 5: Linking FHIR Resources via References [63]

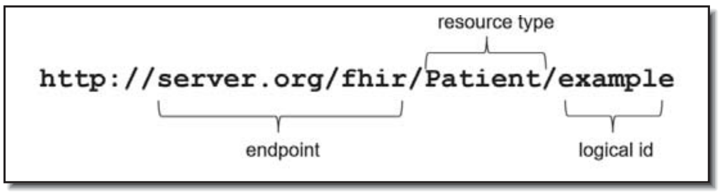


Figure 6: Components of a FHIR Resource Identity [43, p.83]
Licensed by CC BY 3.0

2.2.4. Semantical Interoperability in FHIR

A characteristic aspect of semantical interoperability in FHIR is its use of a complex but structural well-aligned system for communicating the meaning of a specific resource element. The main principles of this idea are based on terminology service functionality. As a result, computer-processable collections of unique identifiers systematically combine concepts or entities with a description, conveying their meaning in a concrete context to minimize variability when analyzing the expressed information in a resource.

An unambiguous interpretation of elements in FHIR is achieved through the usage of the resources depicted in Figure 7. Throughout the specification, elements, like in Observations and Conditions, are defined to include identifying codes which are either defined by the FHIR specification itself, external terminologies[8] or other structured and identifiable enumerations. These collections, referred to by FHIR as CodeSystems, can contain a list of codes and define a compositional syntax through which new expressions for specific health care concepts can be constructed.

For a particular context, it may be more useful to only include a subset of codes from one or more code systems. This composition can be implemented using a ValueSet. This resource allows to pick specific codes, a group of codes based on a filter or include all codes from a specific code system.

[8] In the context of FHIR, the two most prominent terminologies for medical terms or clinical observations, SNOMED CT [66] and LOINC [67], should be highlighted.

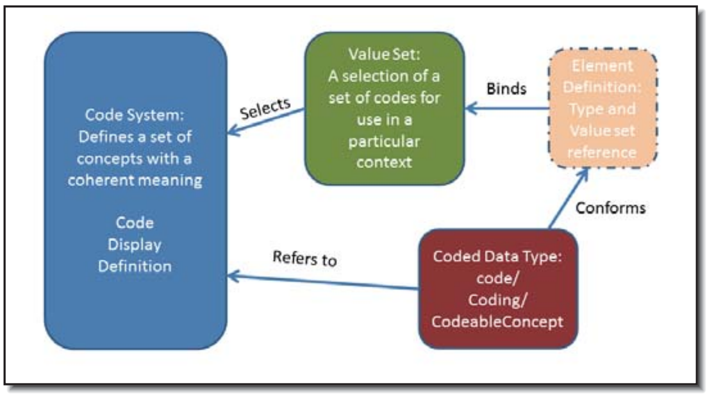

Code System:
Defines a set of concepts with a coherent meaning
Code
Display
Definition
Selects
Value Set:
A selection of a set of codes for use in a particular context
Binds
Element Definition: Type and Value set reference
Conforms
Refers to
Coded Data Type:
code/
Coding/
CodeableConcept

Figure 7: Terminology Resources and their Relationships [68]
Licensed by CC0 1.0 Universal

A ValueSet is related to a Code System, as it defines the include and exclude criteria. It is conceptually relevant to notice that a ValueSet does not define or copy the codes of the referenced CodeSystems. To gain an enumeration of all codes which match the conditions specified in the Value Set, a client can ask a FHIR server to "expand" a Value Set through a $expand REST operation.

Each ValueSet can be identified by a canonical URL, which can be specified via an "url" element in the ValueSet resource. It is expected that this URL does not change and will always point to a "master copy" of this value set, independently where the resource is currently being stored or interpreted. It is to be noticed that this not only applies to ValueSet but all conformance resources, as well as most terminology resources.

Consequently, resources can make use of a ValueSet by "binding" it to a specific resource element. This linkage can be defined through an ElementDefinition which specifies a reference to a ValueSet, a strength indicating whether or not a resource is still conformant if it chooses not to include a code from the suggested/preferred ValueSet, and which of multiple possible Coded Data Types is to be used:

- **Code**
  A single string value from a CodeSystem. The systems from which the codes can be drawn from are only defined implicitly in the definition of the bound ValueSet. This information is not encoded in an instance of the resource as part of the code.
- **Coding**
  A code combined with explicitly stated additional information such as a reference to the used CodeSystem, its version, and a human-readable description of the included concept.
- **CodeableConcept**
  A CodeableConept represents a reference to one or more terminologies by having an enumeration of possibly multiple Codings. Each Coding can describe the referenced concept with a variety of codes from one CodeSystem or refer each time to different ones. Besides describing concepts through Codings, it may also be characterized by a textual representation.

2.2.5. Organizational Interoperability by using FHIR

With regard to its intended scope, FHIR can be categorized as a “platform specification” [69], meaning that it enables the formation of diverse but nevertheless interoperable solutions. The decisive aspect here is that FHIR can be adopted as a foundational layer on an individual basis, even when needing to comply with compulsory requirements regarding the structure of the exchanged information. Building upon broad capabilities for profiling and extensions, an additional layer of interoperability can be achieved. In such a way, FHIR helps to lower organizational barriers, as it can be specified which behaviour is expected and necessary in order to adhere to policies arising from the context of a health care data exchange.

Mandatory constrains to resources may, if needed, be expressed explicitly via different artefacts. Changes to the resources are captured in profiles, which can be associated with an accompanying documentation called Implementation Guide (IG). It supports clarifying the exact usage of a profile. As long as rules of the FHIR specification are not broken or relaxed, resources can be modified, which include, among others, altering the cardinalities of the resource elements, limiting the choices of available data types for a resource element, or specifying the binding strength of a ValueSet. Numerous IGs can be found in the FHIR Implementation Guide Registry [70].

Changes introduced by profiles are outlined in a computable way using a StructureDefinition, whereby it becomes possible to efficiently validate an instance of a resource with regard to the applied profile. This description of how to use a specific resource in a concrete context can be viewed in different forms: profiles can carry only the differences to their base profile (the profile it is derived from) or additionally be completely populated, showing the complete resource structure with all modifications applied.

The challenge of having to accommodate local differences, e.g., on a national, regional or institutional level is addressed in FHIR by allowing to create hierarchies of profiles through re-profiling. Each specialization can be expressed, and commonalities can be inherited from other profiles. This approach highlights the value of sharing and re-using already existing profiles through specially designed registries like Simplifier.net [71].

Interoperability is gained on an abstract level by encouraging collaboration on the issue of creating profiles. Lastly, it also facilitates the accelerated development of FHIR solutions by being able to search for “prior art” and don’t start off from scratch. [72]

2.3. Big Data in the Context of Health Care

The following section contributes to an analysis of challenges which may encounter in the context of Big Data in health care. The presented details are meant to support the rationale for having open and tailored functionality for executing search, query or similar operations in FHIR. Primarily, this line of argumentation is derived from the need of enabling to scale as a standard from small organization-specific health repositories to national health databases regardless of possible dataset sizes and related requirements. Specifically selected health care information must be extractable, as otherwise the full potential of electronic health records (EHRs) cannot be reached.

2.3.1. Definition EHR, PHR, EMR

However, before analyzing Big Data in health care data itself, the form of representation of health records is discussed below. With respect to the scope of health records, it is crucial to differentiate between possible methods for representing a digital version of a patient chart or other health care related information. Selected implementation choices can have a significant influence on how the health records can be used. Unique features, like the fact that user-entered and therefore potentially unreliable data may be present in some kind of these records, needs to be accounted for when performing analytics on top of the health care data.

Similar to the issue of having many isolated and non-interoperable information exchange solutions in health care, there exist a multitude of different definitions of the term "Medical Health Record". Especially, challenges revolve around the issue of defining a clear scope for such a record.

In the current section, an attempt will be made to provide a selection of initial definitions and to highlight important semantical differences. As a result, the thesis aims at providing a differentiation between "Electronic Health Records", "Electronic Medical Record" and "Personal Health Records".

For reasons of preventing misconceptions and ambiguities, standard definitions for the terms listed above are presented. In cases where possible, ISO standards are used to capture a "normative" view on each term. Differences to this approach are indicated at the appropriate place.

> **Electronic Health Record (EHR):**
> "⟨basic generic form⟩ repository of information regarding the health status of a subject of care, in computer processable form". [73]
>
> For more information, please see ISO/TR 20514:2005 - Health informatics - Electronic health record - Definition, scope, and context.
>
> **Personal Health Record (PHR):**
> "health record for which the subject of care or a legal representative of the subject of care is the data controller". [74]
>
> For more information, please see ISO 13606-1:2008 - Health informatics - Electronic health record communication - Part 1: Reference model
>
> **Electronic Medical Record (EMR):**
> "An application environment composed of the clinical data repository, clinical decision support, controlled medical vocabulary, order entry, computerized provider order entry, pharmacy, and clinical documentation applications. This environment supports the patient's electronic medical record across inpatient and outpatient environments, and is used by healthcare practitioners to document, monitor, and manage health care delivery within a care delivery organization (CDO). The data in the EMR is the legal record of what happened to the patient during their encounter at the CDO and is owned by the CDO". [75]

To counteract against a plurality of definitions, the mentioned ISO TR 20514 recognized the need of having a complete set of categories for the term “EHR”, including precise definitions for their respective characteristics. By analyzing the definition of an EHR, it can be noticed that it is defined as an umbrella term to include different specializations. In addition to a basic form, the referenced ISO TR 20514 continues to define multiple subtypes, e.g., shareable / non-sharable EHRs as depicted in Figure 8. Shareable EHRs support use cases involving an exchange of data with external partners and their systems.

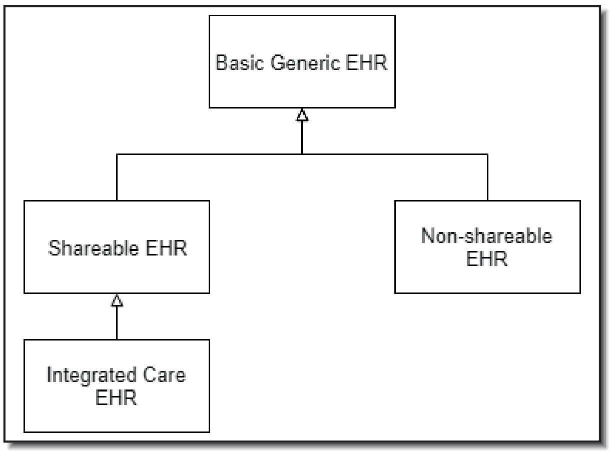


Figure 8: Specialization Hierarchy of EHRs

PHRs and EMRs differ in a subtle way, i.e., can mostly be differentiated on the aspect which entity acts as the data controller of the health records. However, both do not define limitations regarding the scope of its content, at least on the level of their definition. Nevertheless, in practice, a PHR can be more inclusive with regard to information which is not strictly “medical”, e.g., wellness-, nutrition-, or workout-related information. Exemplarily, this assertion can be examined through popular implementations of PHRs, like HealthKit from Apple Inc. running on iOS [76].

EHRs and EMRs are related in the sense that an EHR capture medical information from all practitioners and clinicians involved in the patient’s care. Therefore it is not organization-specific.

The portrayed semantic differences are essential as so far as that all health care records cannot be treated entirely equal. Instead, depending on the use case, distinct operational details need to be recognized. Notably, in the context of FHIR, it is crucial to identify unique characteristics of EHRs. For example, if the assertion would be made that only sharable EHRs would be in the focus of FHIR, potential requirements for the standard’s API may have been added which could result in conflicts for non-shareable EHRs. Therefore having a preliminary “ontology” of EHRs helps to identify unique requirements. Implicitly, FHIR supports the entire spectrum to an extensive degree, due to not marking the REST API as mandatory. Instead, non-shareable EHRs still have the option to just use FHIR resources as their data model internally.

In spite of the presented line of argumentation, it is to be noticed that some authors regard the differentiated terms as synonyms [77].

2.3.2. Discussion: Integration of FHIR in Big Data Systems

Due to a rapid development in the field of Big Data, health care providers are more and more expected to acquire and apply capabilities to improve health care through insights from data analysis. The qualifications which are needed to succeed include the ability to manage (acquire, store, process) a large volume of data, implement data analytics on top of it to gain health-related insights, and integrate these into the whole system such that practitioners, as well as patients, can benefit from the achieved advances [78]. Following a general trend resulting from Big Data in enterprises, this could lead to cost reductions through personalized care plans and the reduction of medical errors through having more information about a patient available from various sources.

The contribution of FHIR and the importance of having a (semi-)structured data[9] format becomes apparent when analyzing the difficulties imposed by unstructured data, which commonly accounts for around 80% of health data [79]. In a wide variety of use cases, medication records or laboratory results often comprise such data. However, the information captured in these free-form texts are even unstructuredly essential as:

- valuable clinical assessments may be included which indicate special conditions of the patient
- the patient, as the owner of the data, has the right to process the data (preferable digitally) to achieve an integrated view of her / his health, depending on the respective legislation

As a result, systems are, regardless of technical difficulties, responsible to gracefully handle even use cases in which the involved data is not presented in an optimal and machine-readable way. In contrast, it is noticeable that when more structured data is available, health care systems can incorporate the information more effectively and efficiently. The cost of dealing with the described heterogeneity reduces, as interoperability lessens the complexity of handling the involved systems.

Still, systems executing the analytics on top of the EHR, should be principally equipped to handle all kinds of data, at least on a fundamental level, as variety is one of the main characteristics of this paradigm, though current developments, as presented in [80], opt for trying to analyze these unstructured information and re-structure by using FHIR.

The FHIR (Search) Framework takes on a special place in this discussion. In numerous proposed architectures for using Big Data in health care, these or related APIs play only a limited role. Alternatively, classical Big Data technology stacks are suggested in literature for the central part of acquiring, transforming and analyzing the data. Exemplary proposals for Big Data architectures can be further explored in [81]. These well-established methods and techniques may be favored due to their specialization and tested reliability, as Big Data is by now a well-studied field in computer science in general. Despite competing with predominant solutions in this field, it does not ultimately mean that there is no possibility for incorporating the FHIR Search Framework into traditional systems or that there is no need for it on a practical level. It is insufficient to label FHIR as fitted to only serve as an underlying technology layer, e.g., by solely treating it as a data format.

[9] The classification of FHIR resources as semi-structured data is vague in some cases as FHIR profiles can outline strict format restrictions, whereby the differences between semi-structured and structured data become indistinguishable.

As an opposing argument, it is valid to indicate that the potential strength of dealing with FHIR, including searching, does not come from its syntax or any other defined capabilities, but from the general possibilities for the interactions and integration it enables. Through its covered levels of interoperability, it empowers a comprehensive integration of information if the standard is understood natively by a system.

Following this logic, it may not be necessary to define a feature-complete search / aggregation / query framework within FHIR while trying to replace already existing technologies. Instead, the standard should “pave the way” for further interoperability by acting as a flexible common denominator applicable for building bridges in cases where needed. For searching, it would result in the possibility of enabling the execution of online searches on FHIR resources based on their structural properties. This capability can, in turn, be used in any use case where appropriate, while relying on similar search behaviour across systems.

Supporting this argument, an example of a more advanced cloud-based analysis of health care data using a FHIR-enabled Big Data solution can be found in [82]. These kinds of systems can be used to enable the reasoning about the involved health care data, focusing on their primary goals and strengths while outsourcing complexity to FHIR.

On a theoretical level, an alignment towards such the described strategy can be derived from the fundamental FHIR design principles presented in section 1.3. The discussed matter is important in the context of this thesis to the extent that non-coordinated developments in the standard may be detrimental, e.g., by continuously inflating the scope of it or adding complexity for implementations while losing focus of its core aspects. Mostly, new additions towards querying, opposed to searching, would fall into this category.

Throughout the following chapters, the hypothesis that support for a certain capability may be compensated through a “FHIR-compatible” third-party solution is to be evaluated based on the example of discussing the implementation of aggregation, querying, and filtering.

Please note that all arguments presented in this section hold true for other equivalent standards besides FHIR as well. FHIR has only been selected as it is expected to gain traction in the future and provide through its distribution a reasonable incentive for implementation.

2.3.3. EHR Statistics Netherlands

The following section describes an initial approach to validate the issues resulting from Big Data in EHRs. To comply with imposed requirements resulting from managing EHR information, statistics from a selection of university hospitals in the Netherlands are provided within the next section. Based on these data sets it is analyzed to which degree the V’s of Big Data[10] have an influence on real-world applications.

Due to the sensitive nature of the involved information, data could only be gathered from a small number of EHRs for the publication in this thesis. As a result, concrete conclusions from the data cannot be generalized. Nevertheless, the findings may serve as an initial indicator of resulting challenges.

[10] The reader is referred to [83] for a detailed introduction to Big Data.

With the intention of underlining the need of having proper support for searching and related methods in FHIR, as discussed above, EHRs have been selected for illustration purposes. They are expected to contain the most extensive set of health care information, thus it is hypothesized that more difficile requirements exist than, for example, in PHRs due to their broader scope.

All of the following examples are gathered from statistics made available by the VU university medical center Amsterdam (VUmc) and the Leiden University Medical Center (LUMC). All original information is provided in Appendix B.

Figure 9 and Figure 10 depict an aggregated view of the data space used (in KB) per instance of a so-called Health and Care Information models (HCIM), also known as Detailed Clinical Model (DCM). These terms refer to logical models which provide, similar to FHIR resources, data elements for capturing different aspects of a health care domain. For example, the "DCM_Afspraak" can be used within the VUmc to store appointment related information. In the Netherlands, these models are designed as "building blocks" and managed by Nictiz, the center of expertise for E-Health [84]. HCIMs can be used to work towards basic content-level interoperability in order to reuse the captured information throughout the health care process (e.g., from the use within EHRs to patient-related research). The the mentioned figures, each bar in the charts (from left to right) correspond to a HCIM, labeled as a DCM, in Appendix B (from top to bottom).

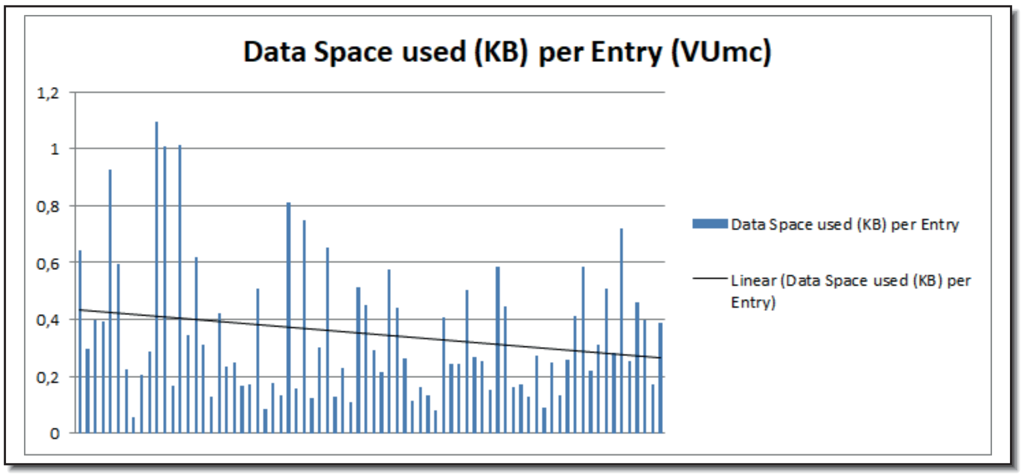


Figure 9: Data Space Used (KB) per Entry (VUmc)

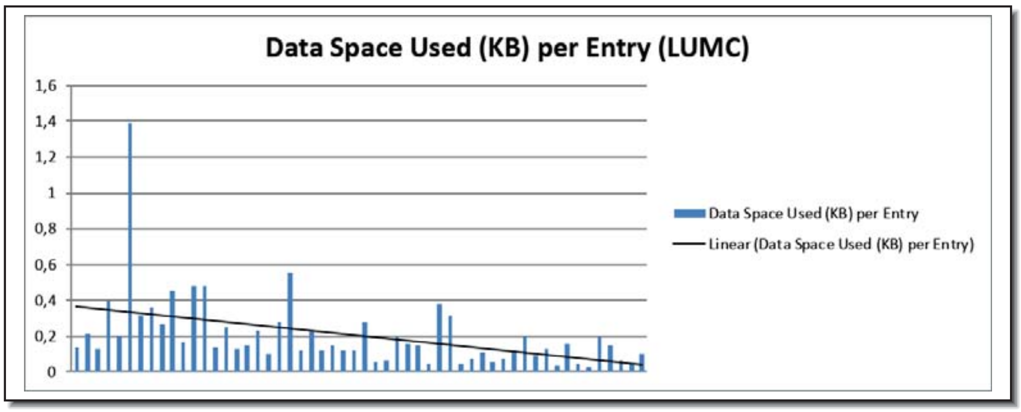


Figure 10: Data Space Used (KB) per Entry (LUMC)

Interestingly, in the statistics above, the ratio of Data space used (KB) / Number of entries is relatively low. For a row count of 3.436.822.051 (LUMC) and 78.459.012 (VUmc), an average of approximately 0.35KB is used. The relation of size per entry does not exceed 1.5KB. Regardless of the timeframe in which this data has been gathered, it is to be noticed that the quantity (97.43GB in total at VUmc, 780.9GB in total at LUMC) does not reach a limit at which it is to be categorized as “big”. Excluded in these data sets were personal identifiable information, such as a BSN (Citizen Service Number), images (due to clinical workflows managed in a separate DICOM-enabled system) and genetic information.

Such a finding can indicate that, as discussed throughout this chapter, interoperability is a significantly bigger technical challenge than data management in an EHR. This hypothesis is underlined by the Figures 11 - 13. In the Vumc, LUMC and the University Medical Center Utrecht (UMCU), official HCIMs provisioned by Nictiz accounted for 35% on average for the used data models. All other data models which were used in operation were custom designed. Interoperability can than only be established through mappings of HCIMs.

In Appendix C an attempt is made to provide an overview of HCIMs which can be roughly mapped between each other. The included tables also provide a mapping from HCIMs to FHIR resources for some of the models. These resource mappings are officially created by Nictiz in an effort to foster FHIR usage via the MedMij project [85].

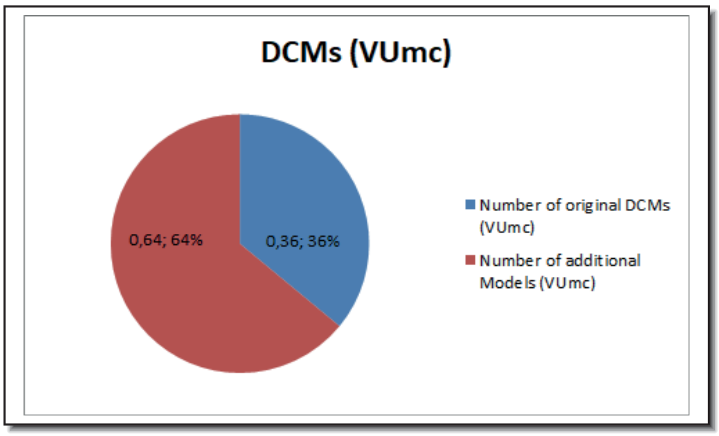


Figure 11: Comparision Original / Custom DCMs (VUmc)

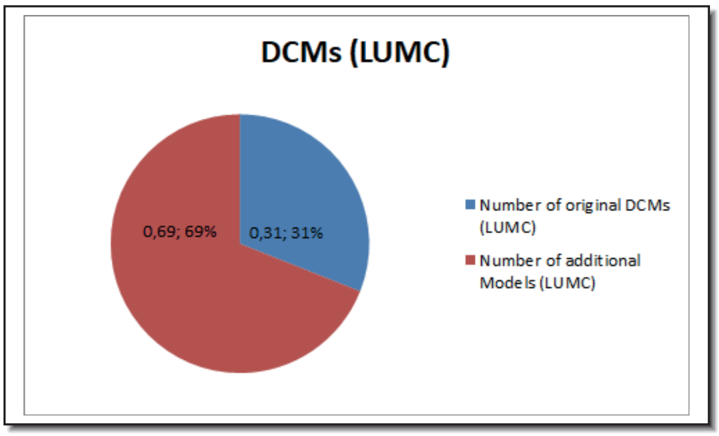


Figure 12: Comparision Original / Custom DCMs (LUMC)

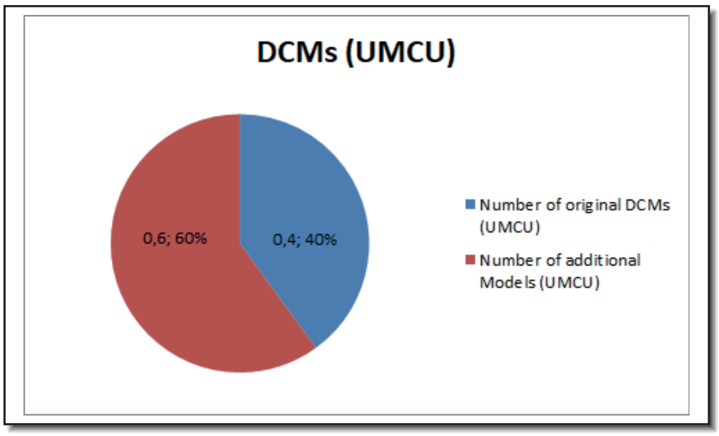


Figure 13: Comparision Original / Custom DCMs (UMCU)

2.4. Essential Definitions

Lastly, to conclude the theoretical background, a selection of essential conceptualizations is provided. By specifying the meaning of these concepts within this thesis, a detailed understanding of frequently emerging key terms is to be reached. The subsequently discussed concepts have been chosen, as they are often used in similar contexts and are therefore subject to confusion and misinterpretation. Hence, some of the concepts pose an increased risk of being erroneously used as synonyms.

- **Querying**
  Querying can be interpreted as retrieving an exact collection of information from data sets by specifying an information need[11] precisely in a formalized language [86]. Such a request is handled by interpreting the resulting query and returning a set of matching information. Therefore, a query can be defined as a function [87, p.56] that, when applied, returns a selection of relations, objects or documents, depending on the underlying data model (e.g., relational, document-based or graph models). By having the only restriction of conforming to the syntax of the query language, the information can often be requested in a quite flexible way.
- **Searching**

  Searching refers to the approach of roughly defining an information need through the usage of the restricted parameters and operators. Based on this combination, queries can be derived which in turn retrieve the wanted information by traversing the data sets and selecting matching data entries. Resulting from this definition, the difference to querying is the expressiveness of the methods and their level of abstraction. Searching consequently enables users to freely explore a predetermined / indexed range of data sets [88].

  A comprehensive introduction to the search mechanisms of FHIR can be found in section 3.1.
- **Filtering**
  In contrast to searching, filtering is based on the idea of reducing a preliminary result set by applying filter criteria to it. These criteria can correspond to the same parameters and operators used for searching. However, instead of building a result set, it is minimized to only include specialized information. [89]
- **Aggregation**
  In the case of querying specific data sets, most commonly the outcome is returned in the form of a result set. This result set is being gathered by evaluating the expression of the query statement for each data entry (e.g., each tuple in a relational database, or each document in a document-based database). However, it exists the need to reduce these sets by applying an aggregate function to it. Examples comprise the following: count(), sum(), max(), min(), avg(), StdDev(). Through the application of these function, a condensed version of the result set is given, consisting of a combination of the original values. [87, p.113]

[11] This term is defined precisely and discussed in section 4.1. For the comprehension of the current section, its literal meaning should be identifiable through the context.

3. Description FHIR Search Framework

The intention of the following section is to provide a structured overview of the functionality encapsulated within the FHIR Search Framework, which can be seen as an extension to the REST API described in section 2.2.2. It defines additional API endpoints for searching in FHIR resources. In consideration of the preceding definitions of searching and querying, it should be noted that the defined API interactions and other related concepts are solely meant, as the name of the framework suggests, for searching. Besides having limited support for performing requests in a fine-granular way, as subsequently demonstrated, the specification does not define a "feature-complete" server-side querying framework. Specifying such a capability is still subject to research and is listed as an outstanding issue of FHIR [90].

The FHIR Search Framework rather seeks the goal of specifying how to traverse resources and explicitly search for information that match provided search criteria. Figure 14 depicts the individual components through which search requests can be formulated. The following sections aim at describing how these components can be used in combination and which semantical meaning they express. A complete specification of the search requests' syntactical structure using a formal grammar is not given. Details about the exact construction of the search request URLs are subject to the FHIR specification [91] and out of the scope of this thesis. It is argued that the semantical meaning of the search requests are more stable than its syntax. Therefore a description of the search request's meaning is seen as more informative. The following description of the search functionality is all based on the last mentioned specification reference. Examples for search requests and an explanation of their meaning are provided in Appendix D.

With respect to a semantical analysis of the FHIR Search Framework, a model for abstractly describing a search request is proposed, based on the ideas formulated by Kim, Lee, and Choi regarding the representation of a user's search intent [92]. The pursued aim is to achieve a way of dissecting the (potentially complex) search requests into their individual components, which would allow an analysis of how they are logically connected. Having such a description within this thesis is seen as a basic prerequisite for analyzing the limitation of the FHIR Search Framework afterwards. Without such a model, only a limited foundation for reliably assessing the full extent of the framework would exist, due to the fact that it would not be directly discoverable which search components can be used together. Moreover, such a description would contribute to helping a user of the FHIR Search Framework analyze what semantics a search request has, as all implications of this request can be presented in a structured and precise way.

In each of the next sections of this chapter, a specific part of a search request is discussed. As a result of this discussion, it can be shown that every search request follows a cohesive structure. Section 3.6 provides a summary of this analysis by presenting the complete model of a search request. It is to be clarified that the following descriptions are not to be mixed up with a critical discussion of effectiveness or efficiency of each component or the framework itself. This assessment can be found in chapter 4 and 5.

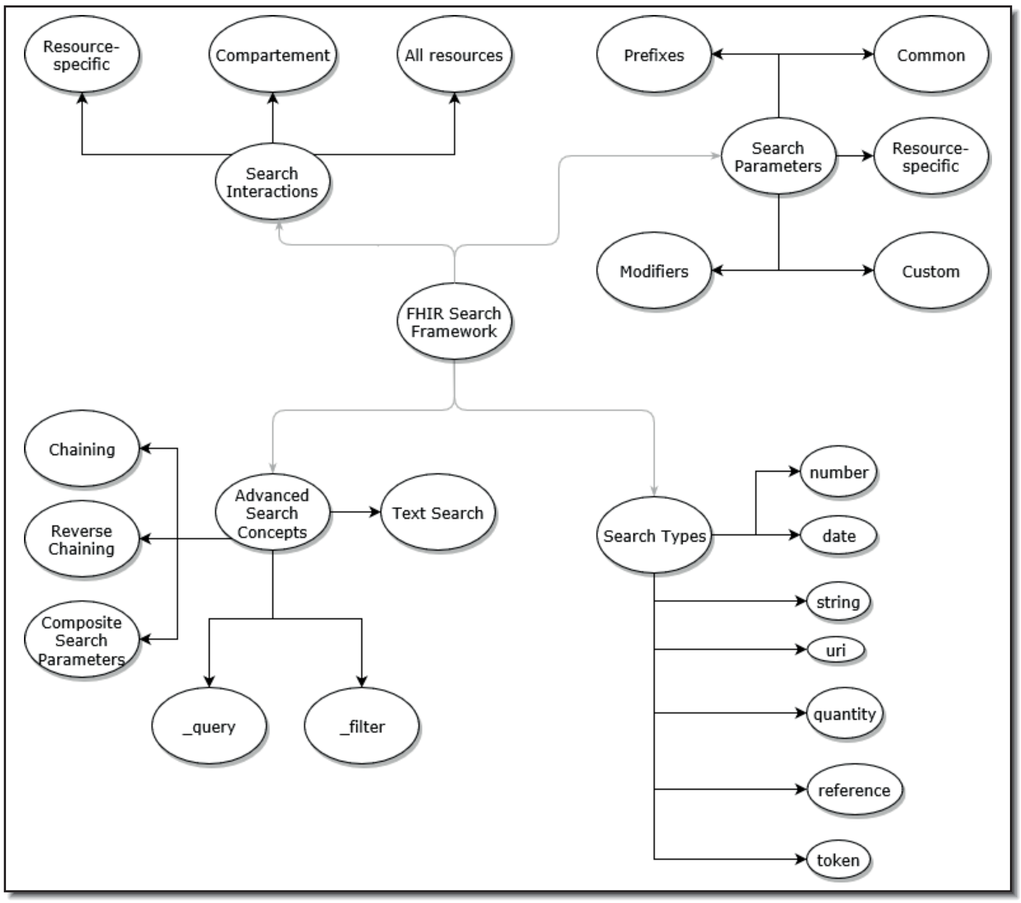


Figure 14: FHIR Search Framework Components

3.1. RESTful API: Search Interaction

The search interaction defined by FHIR is on a fundamental level a request containing pairs of named search parameters and corresponding values, whereby each search parameter is logically referring to one or more resource elements. The exact mechanisms of how such a mapping is achieved are presented in section 3.2. On the basis of these search parameters, it is evaluated by the receiving server which currently stored resources match the given criteria, i.e., which resource elements correspond to the search parameter values. A set of matching resources is subsequently returned via a bundle of type “searchset”.

Each search request is directed by a requesting client to a specific search context which shall be searched. Each of the following contexts defines their own scope of which resources are to be considered for a search match:

- **Search all resources**
  Certain parameters are common to all resources and can therefore serve as the foundation for a search, which is directly addressed to the base URL of the FHIR server. By this means, the narrative and resource metadata of the resources, i.e., the information that the majority of resources contain as common parameters, can be explored.

- **Search a selection of resources**
  By using the search parameter _type, the standard allows a search request to be forwarded to a selectable collection of resources. In this case, an enumeration of resource types is provided followed by the search parameters, which shall be used as a filter. This approach is applicable in cases where not all resources are to be searched regarding their common parameters.

  The specification additionally adds flexibility by allowing search parameters that are not common to all resources to be included in this kind of search. As a result, it is possible to search for parameters that are only included in a subset of resources, e.g., a search on StructureDefinition and ValueSet that include a specific part of a canonical URL, as both contain the resource element "url".

- **Search a compartment**
  In the FHIR specification, resources are presented in arranged logical groups, combining resources that are similar in their meaning, i.e., share the same overall concept. For example are Patient, Practitioner, and RelatedPerson all grouped as Individuals. Along with this approach, FHIR also defines groups that are not based on the inherent meaning of the resources but their content-related elements. Specific resources that often share explicit references with other resources form a compartment [93]. These logical groups combine all resources having any outgoing references to their shared target. The FHIR Search Framework defines in these cases a way of efficiently searching these coupled resources by allowing to search a specific compartment, without needing to state the resources that it comprises.

  These search requests match any resources that have any outgoing reference to the resources on which the compartment is based. For example, when searching the Patient compartment it would be possible to match any AllergyIntolerance that has any link to a patient (Patient with [id]) via either its patient, recorder or asserter reference, and that fulfills the search criteria.

- **Search a specific resource**
  FHIR allows, in its simplest form, a search on one single resource based on a AND combination of one or more search parameters. As a result, it matches all resources that conform to all search parameters. Although, it is possible for an executing FHIR server to include references to resources that do not precisely match but are still believed to be relevant to the current search context. Moreover, it is possible to realize simple OR searches by providing multiple values per search parameter.

Please note that for each discussed syntactical component of a search request, there is a detailed example provided in Appendix D. In these examples, some elements like [base] and <resource element> are meant to be interpreted as a variable. These examples are not valid without replacement. This notation is used in throughout all examples. Furthermore, these examples are only to be conceived as such. In some instances, more complex cases and optional additions exist, which are not covered in the listing presented here. As stated above, the exact allowed syntax is subject to the FHIR specification in its current version.

Listing 8 up to Listing 10 present examples for the discussed search requests in this section.

3.2. Search Parameters

In general, the following options represent valid search parameters in addition to the common search parameters, which have been illustrated in section 3.1. All search parameters can be used in the search interactions presented above to request resources by restricting the search result set.

- **Resource-specific Search Parameters**
  Following the theme of having computable definitions, FHIR provides a way to structurally define search parameters through a SearchParameter resource. One of the resource's purpose is to uniquely identify the search parameter in the form a canonical URL. An explicit identification attribute and a chosen name combined with a computable expression, describing to which resource elements this search parameter applies, establishes semantical interoperability.

  The unambiguous selection of the right resource element(s) is effectively achieved across systems by specifying the desired path within the resource in a so-called FHIRPath expression. FHIRPath is a language for traversing a directed acyclic graph and enables the selection and filtering of elements along a given path. As FHIR resources can be abstracted from its serialization form and mapped to such a data model [94], FHIRPath can be used in the definition of search parameters.

  Each FHIR server can specify within its CapabilityStatement which search parameters are supported. This selection can be freely chosen with regard to the clients systems' needs. Per resource, specific search parameters are defined and distributed by the FHIR standard to achieve interoperability on an organizational level. For this reason, the usage of the same search parameters across systems is facilitated by pursuing the same 80/20 principle, which applies for the selection of resource elements [95].

- **Custom Search Parameters**
  If more flexibility compared to the already defined search parameters is needed in specific cases, servers may choose to support custom search parameters. Interoperability can, however, be retained by including the custom search parameters in the server's CapabilityStatement, effectively allowing dynamic discoverability at a minimum level. The meaning of the search parameter is, same as discussed above, describable through a SearchParameter resource, which can be published and provided with an arbitrary but stable canonical URL. To avoid (backward-)compatibility issues, care should be taken when naming search parameters, such that they do not clash with names defined by the current or future version of FHIR. Currently identifiable naming schemata, which are partially derivable from the resource element type or name, should be taken into consideration.

Listing 11 presents examples for the discussed search requests in this section.

3.3. Search Parameter Types

In the most trivial case, a search request matches resources by checking the equality between the search parameter values and the information stored on the enquired server. The definition of how to check if a resource element matches, depends on special factors. Each search parameter is per definition associated with a search parameter type, which is verifiable by accessing the resource element "type" in each SearchDefinition. Individual types define for themselves by which means a correct and interoperable behaviour across systems can be insured.

For example, when searching on a resource element that could be expressed as a number, equality is defined with a custom precision range depending on how the search intent is expressed. If the client indicates that high precision is needed by including fractional digits in the number, servers are expected to evaluate a number up to five significant figures. In all other cases, three significant figures are sufficient. Other specialized behaviour is defined across all search parameter types to take individual structures and requirements into account.

The search parameter types, technically defined in a search-param-type CodeSystem [96], are independent from the FHIR Data Types [97] used in the StructureDefinitions of each resource, i.e., the data types of the search parameters' target(s). By aggregating resource data types into their respective search parameter types, commonalities are leveraged to provide an API that offers an effective search across a multitude of complex and primitive types but still retain implementability.

Most of the search parameter types only define limited variations regarding the structure of valid search parameter values. Especially for number, string, and date search parameters, a match is in most basic cases only determined based on a single corresponding value. However, there exist a few search parameter types that are needed to search on resource element types that are special to FHIR and for which complex search parameter values may need to be specified. In particular, special rules for how to process these combined values are needed to search on token, reference, and quantity search parameters.

- **Token**
  To search on a resource element which can contain a code in any way (e.g., through having code, Coding or CodeableConcept as a resource element type), the search parameter value may contain a code and additionally the CodeSystem in which the code is expected to be specified. Moreover, any combination of these two values is possible for a search value. Through specifying a valid combination it is possible to:
  - Match a code independent of a specific CodeSystem
  - Match a code where no CodeSystem is specified in the resource
  - Match any code in a given CodeSystem
  - Match a specific code and CodeSystem

- **Reference**
  When using certain search parameters which have a narrow search scope, it is possible to infer the resource type of a search parameter's target. Based on the search parameters target FHIRPath expression, the resource type can be determined. Disjunctions of multiple types are allowed to create cross-resource search parameters, even for references. In cases where the search parameter is used to search on a reference and the type can be inferred, clients may choose to omit the type and provide only the logical id of the resource that was searched for as the search parameter value. In all other cases, search parameter values for references can contain the type of the target resource and a logical id. This logical id is evaluated within the scope of the server which is currently being searched. However, it is also possible to search based on an absolute URL. In the latter case, references to local and external resources can be matched.

- **Quantity**
  To achieve interoperability in health care, it is necessary to ensure that observations, such as laboratory results, are encoded using standardized units (e.g., measuring an HbA1c value for diabetes-related purposes using mmol/mol). Only by this means it can be ensured that different systems can write and retrieve the observation values and understand, as well as process them in a meaningful way.

  However, such a precise documentation is not always given due to different organizational factors and a missing harmonization of units for recording purposes. Consequently, a recorded measurement may be specified ambiguously. The authors of [98] present the importance of this issue by showing examples where up to 68 different descriptions for a single kind of measurement could be found in clinical laboratory test results data tables.

  To prevent such issues, FHIR states that wherever possible, a quantity should be encoded using a code from the "Unified Code for Units of Measure" CodeSystem (UCUM). The main goal of UCUM is to provide semantically and computable definitions of "[...] all units of measures being contemporarily used in international science, engineering, and business" [99]. This aim is tried to be fulfilled by defining a formal grammar which enables the generation of human- and machine-readable codes with a precise and comparable meaning.

  On a technical level, UCUM defines at first a set of so-called base units. For this purpose, the metric units meter, second, gram, radian, kelvin, coulomb, and candela are proposed. These base units can be represented as a set of a standard basis vectors $B = \{b_1, b_2, ..., b_n\}$. Each of these vectors has a dimension which is equal to the number of base units, e.g., the basis vector for meter would be $(1, 0, 0, 0, 0, 0, 0)$. Please note, that any other set B' can be used as base units as longs B' is isomorphic to B, meaning that each unit from B' can be transformed into one and only one unit of B by a bijective mapping. In summary, every unit can be a base unit if it is mutually independent of any other unit and it can be transformed into the UCUM base units using a linear transformation. For example, if for any reason it would be more appropriate to select a kilogram as a base unit, specifying it would be possible, as kilogram can be mapped to gram. For more details about the base units see [99, §28].

After having defined the base units, it is possible to reduce any other expression of a unit to its canonical form, i.e., its form where the unit is expressed using only base units. Such a reduction is possible as any unit needed to represent a measurement can be represented through the following operations:

- a multiplication between units
- a division between units
- a exponentiation between units
- multiplication of a unit with a scalar

As stated in [99, §20], it is possible to transform any conventional unit into a canonical form in the form of a pair $(r, \hat{u})$ with $\hat{u} = r * (u_1, u_2, ..., u_n)$ where u_i is the dimension of the respective base unit, i.e., its exponent and r is an arbitrary scalar factor which may be needed for alignment purposes.

For each of these mathematical operations, UCUM additionally defines a set of symbolic operators which can be combined. Therefore, any string which conforms to UCUM represents a unit which in turn can be transformed to its canonical form. A unit is valid if it can be generated by a set of formal production rules defined by UCUM, which are verifiable using a pushdown-state automaton [99, §10]. Through these mechanisms, a machine-interpretable grammar and an algorithmic way for transforming any unit to its canonical form is provided by the specification.

Transformations of units to their canonical form can exemplarily be found in Table 1.

Example Description	Unit expression	$\hat{u}$
Area measured in square meters	m2	(2,0,0,0,0,0,0)
Velocity measured in meters per second	m/s	(1,-1,0,0,0,0,0)
Volume measured in litre	mm.m2	$10^{-3} * (3, 0, 0, 0, 0, 0, 0)$
Force measured in Newton	1.kg.m/s2	$10^{3} * (1, -2, 1, 0, 0, 0, 0)$

Table 1: Transformation UCUM Units ⇔ Canonical Form

The FHIR specification leverages the fact that canonical forms can be computed based on a valid UCUM expression. At the discretion of the server, search requests may be based on base units regardless of how they were recorded in the Observation resources. A search based on UCUM can be indicated by a requesting client by including UCUM as the code system and providing a valid unit expression afterwards. All expressions which share the same canonical form are regarded as equal and return hereupon.

Listing 12 up to Listing 14 presents examples of the discussed search requests in this section.

3.4. Modifiers and Prefixes

Modifiers and prefixes can be used in combination with a search parameter value to either change which relational operator is used to gather the search result set (prefixes) or change how the supplied value is to be interpreted by the executing server (modifiers).

Search Parameter Type	Modifier
All search types	:missing
references	:[type]
uri	:below, :above
string	:contains, :text, :exact
token	:text, :not, :above, :below, :in, :not-in
quantity	n/a
number	”
date	”

Table 2: Mapping Search Parameter Types / Modifiers

Each modifier is supplied as a suffix to the search parameter name and can change the behaviour of the search parameter as discussed below. A mapping depicting which modifiers are allowed for which search parameter types can be found in Table 2.

- :missing matches only if the search parameter's target is not existent within an instance of a resource.
- :[type] clarifies which resource type is meant when traversing over a resource element that can reference multiple other resources.
- The :below and :above modifier are both defined for URI and token search parameters. Due to their semantic differences, the modifier usage will be discussed separately for each search parameter type.

  For URLs, the :below modifier can be included in a search request to indicate that an element inside a resource which is of type URI is to be evaluated only partially. It is evaluated if the URL starts with the given value and is continued arbitrarily. This modifier can be useful when testing the ownership of a canonical URL. By using the :below modifier, a client can ask to check if a canonical URL is for example within the official realm of HL7 ("below" http://hl7.org/fhir/).

  In combination with the described modifier, :above is likewise defined for URLs. However, no search request that would include this modifier and represent a meaningful semantical statement could be determined.

  A token search parameter matches when used in combination with :below if the code which is submitted as the search parameter value subsumes the code which is stored in a matching resource. The concept of subsumption is explained in Figure 15.

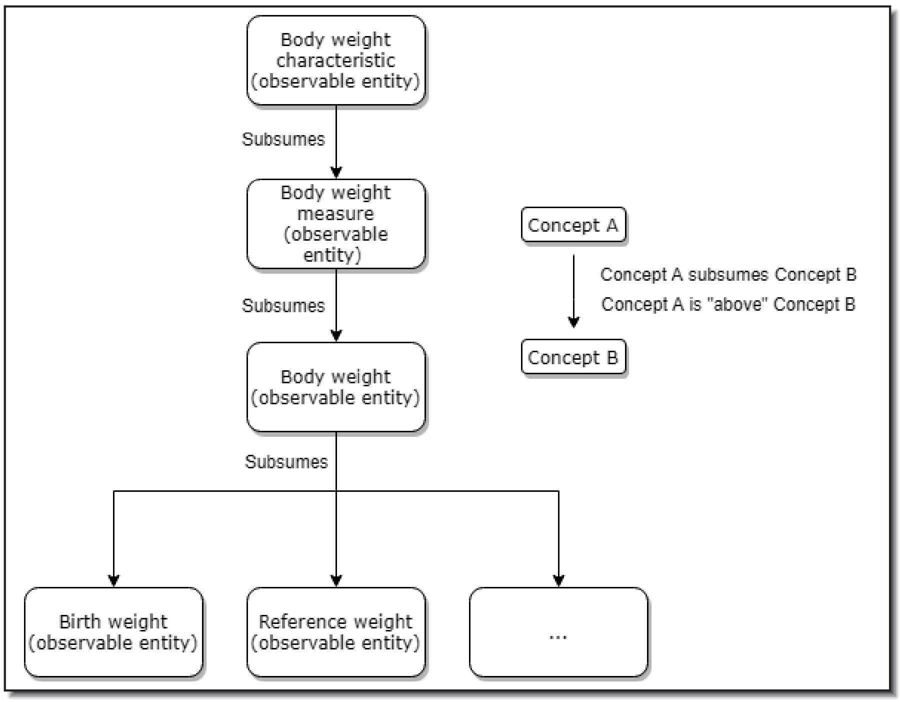

Body weight characteristic (observable entity)
Subsumes
Body weight measure (observable entity)
Subsumes
Body weight (observable entity)
Subsumes
Birth weight (observable entity)
Reference weight (observable entity)
...
Concept A
Concept A subsumes Concept B
Concept A is "above" Concept B
Concept B

Figure 15: Subsumption in Code Hierarchies (Example: SNOMED CT - 363804004)

When given a code which is drawn from a hierarchical Code System :below enables the search for any code below a certain level. Nevertheless, it should be noted that FHIR defines furthermore a "codesystem-subsumes"-Extension [100] which permits to explicitly state subsumption for a code in relation to another in the event that the used Code System is not hierarchical.

The :above modifier can be used in combination with any child node of a Code System. It retrieves all resources that contain parent codes, i.e., codes which are above the given code in the Code System.

- :contains indicates that all resource elements that partially consist of a specific substring at any position in any combination with regard to its casing should be matched.
- :text searches the text portion of a Coded Data Type instead of the code itself.
- :exact restricts the search to exclusively match strings that are identical.

This modifier changes the normally expected behaviour, as a search for a string parameter operates per default, i.e., without the :exact modifier, on an aligned sequence of Unicode characters. The handling of issues related to encoding and character representation is as a result of this simplified by mandating the indexing of a stripped down version of the search parameter value. This version does not contain any combining characters, like accents or other diacritical marks.

As a result of this thesis, it was noticed that certain issues arise from the definition of string identity. Concretely, the specification does not give clear guidance on how to handle so-called (extended) grapheme clusters [101]. These basic units of text represent a single user-perceived character through multiple Unicode code points instead of a single one.

However, both versions are, when rendered, visually not distinguishable. The Unicode specification defines that both versions should be treated as equivalent [102]. On the contrary, some programming languages define the equal operation for a string based on the exact used code points. A consensus regarding this issue in the context of FHIR, i.e. what should be matched by :exact is still to be reached.

- :not matches the set that is the exact complement of codes as specified in the search parameter value. It, therefore, matches all elements that do not contain the specified code or no code at all.
- :in can be applied on code search parameters and retrieves all elements that include a code that is part in a ValueSet. The value set is specified by an absolute or relative URL given as the search parameter value.
- :not-in is defined as the exact opposite of :in. Otherwise, it is to be used in the same use cases and exhibits the behaviour.

The prefixes depicted in Table 3 can be used to indicate that a search parameter value is not to be evaluated based on an equal operation, but some other relational operator.

Search Parameter Type	Prefixes
All search types	n/a
references	”
uri	”
string	”
token	”
quantity	eq, ne, gt, lt, ge, le, sa, eb, ap
number	”
date	”

Table 3: Mapping Search Parameter Types / Prefixes

Listing 15 up to Listing 19 presents examples of the discussed search requests in this section.

3.5. Advanced Search Concepts

As a general principle, the FHIR Search Framework does not mandate a baseline of functionality, which must be implemented by a server. Instead, it allows each server to choose its own scope and announce their implemented features through its CapabilityStatement. As demonstrated in the given examples in Appendix D, major parts of FHIR resources can be accessed through the presented search mechanisms. However, due to the distributed nature of FHIR, it may be a requirement to not only check if a resource does contain a certain reference, but also to actively use such information to introduce an expression for “connectivity”. Based on this, FHIR resources could be viewed as a graph and not only be searched as a list of resources. The FHIR Search Framework recognized this need and introduced advanced search concepts which allow more expressiveness. These auxiliary concepts are described in detail in the following sections.

3.5.1. Chaining & Reverse Chaining

The FHIR Search Framework offers the capability to search on logical paths in resources via chaining and reverse chaining requests. This feature allows building an implicit graph of references and check if it is traversable in a particular direction. Examples include the following requests:

- Get all resources which have incoming references from an Observation to a Patient
- Get all Encounters which are referenced by an Observation
- Get all Patients which are managed by a certain organization.

Concrete chaining and reverse chaining examples, including a description of their semantical meaning, can be found in Listing 21 and 22.

Chaining essentially allows a client to traverse over a sequence of references by "looking ahead". Each "hop" in a chaining request is a reference search parameter and matches if a resource on which the reference parameter is defined is connected to the resource containing the predecessor parameter. After having established a set of resources which includes the given chain of references, such set is further reduced by adding a standard search expression at the end of the chain. A visualization of the described search request can be found in Figure 16.

Chaining is to be categorized as an advanced search concept due to is expressive power. It presents the possibility of searching on arbitrarily connected resources, without needing to know specific resource IDs.

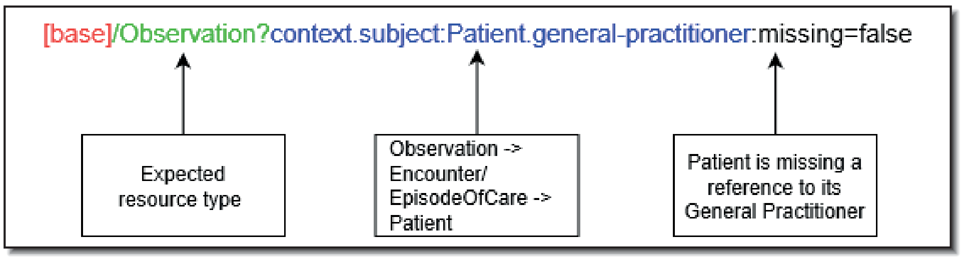

[base]/Observation?context.subject:Patient.general-practitioner:missing=false
Expected resource type
Observation -> Encounter/ EpisodeOfCare -> Patient
Patient is missing a reference to its General Practitioner

Figure 16: Structure Chaining Request

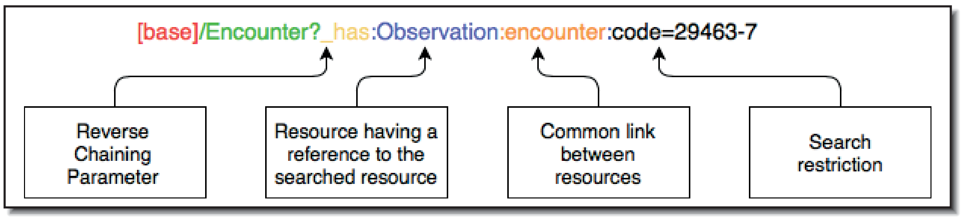

[base]/Encounter?_has:Observation:encounter:code=29463-7
Reverse Chaining Parameter
Resource having a reference to the searched resource
Common link between resources
Search restriction

Figure 17: Structure Reverse Chaining Request

Reverse chaining is similar to chaining as the name suggests. It allows matching resources based on incoming references. Clients can use reverse chaining to analyze if a sequence of common references resources exists which ends at the enquired resource. A search expression for an element defined on the target of the reverse chaining request can be provided, similar to chaining, at the end of the search request.

A visualization of a reverse chaining request can be found in Figure 17. Please note, that although ":" characters are used to separate certain components of the query, they are not technically meant to be a modifier. Only in cases where the destination resource of a reference is ambiguous, a :[type] modifier can be used. In every other case, ":" indicates through which common reference elements two resources are linked in a chain.

3.5.2. Composite Search Parameters

In section 3.1 it was partially described that OR searches are allowed by passing multiple comma-separated values to a search parameter. This capability is however accompanied by the restriction of only having the possibility to express logical ORs for one search parameter. Multiple search parameters can still only be combined through an AND operator[12]. Without additions, this restriction would leave clients of the FHIR Search Framework with difficulties expressing combinations of related resource elements. This issue would manifest itself as described below.

In instances where it would be logically and semantically reasonable to search on values which are strongly connected (e.g., the code and value of an Observation), FHIR provides so-called "Composite" search parameters. These search parameters match resources based on pairs of search parameter values.

Examples of Composite search parameters can be found in listing 20. Each Composite search parameter contains at least two values which are separated using a "$" sign. Each of these values represents a search based on another defined parameter. As a result, it becomes possible to match key/value pairs in resources.

Please note that in some cases the behaviour of a composite search parameter is not different than a simple search which includes the key and value as distinct search parameters. In the example described above both mentioned resource elements have a standard cardinality with a maximum of one. Therefore the search using "?code=<code>&value=<value>" would result in the same result set as "?code-value-quantity=<code>$<value>". This assumption does not hold true for all instances. Whenever a matched resource element has a cardinality greater one, false positives can be introduced in the result set, as no coherence of the parameters is enforced. As a result, elements can be included which do contain both search parameter values but not necessarily as a pair.

3.5.3. Advanced filtering

In addition to the mechanisms described in the sections above, another viable solution for expressing a search request is to use the _filter parameter. This parameter is part of the FHIR Search Framework, besides being mainly defined in a separate section of the specification [103]. Its main purpose is to allow the possibility of combining search parameters of each resource type with logical operators (logical AND, OR, NOT operators).

[12] Expressed in the search request URL through "&" between the individual search parameters.

On an abstract level, a filter expression can consist of three different components:

- A filter can be a logical expression in the form of "<filter> and <filter> | <filter> or <filter>"
- A filter can be a negation in the form of "not <filter>"
- In a filter, parentheses can be used to form logical groups and define the precedence of the operations.

The concrete substitution for a filter always follows the form of

"<paramPath> SP <compareOp> SP <compValue>"

<paramPath> is the name of a search parameter defined for the enquired FHIR resource. Reference search parameters can be chained using a dot-notation. <compareOP> defines on which grounds a filter matches a resource. For this purpose, special operators are defined by _filter. A comparison of how these operators related to the operators used in a "traditional" search request can be found in Table 4. If a interchangeable operator exist in both parts of the specification, the corresponding cell is marked as transformable. In all other cases, it is highlighted where gaps exist. If cases exist where a certain operator would not be applicable, the comparison is marked as such. Depending on the type of the search parameter, a suitable value is to be passed to the filter expression using <compValue>. SP is a white space separator.

A further addition of _filter is the possibility of using aliases for commonly used terminologies, e.g., "snomed" instead of "http://snomed.info/sct".

Operation	String	Number	Date	Token	Reference	Quantity
eq	Default =	Default =	Default =	Default =	n/a	Default =
ne	Undefined	Undefined	Undefined	:not	n/a	Undefined
co	:contains	Default = (Implicit Range)	Default = (Implicit Range)	n/a	n/a	n/a
sw	Undefined	n/a	n/a	n/a	n/a	n/a
ew	Undefined	n/a	n/a	n/a	n/a	n/a
gt / lt ge / le	Undefined	Prefixes	Prefixes	n/a	n/a	Prefixes
pr	:missing	:missing	:missing	:missing	:missing	:missing
po	n/a	n/a	Undefined	n/a	n/a	n/a
ss	n/a	n/a	n/a	:below	n/a	n/a
sb	n/a	n/a	n/a	:above	n/a	n/a
in	n/a	n/a	n/a	:in	n/a	n/a
re	n/a	n/a	n/a	n/a	Default =	n/a

Table 4: Comparison _filter and "Classical" FHIR Search

The highlighted limitations in Table 4 (marked in red) result from the following limitations:

- Only token search parameters can normally be negated. Using _filter, every <paramPath> can be followed by a Ne operator.
- Sw (starts with) operators cannot be replicated using other parts of the FHIR Search Framework. The :contains modifier matches a string regardless of its position inside the search text. The same issue occurs with ew (ends with).
- Only _filter defines a lexicographical comparison of strings using relational operators.
- Similiar to sw and ew, po is not translatable to a similar search request as a whole.; Po checks for overlapping date periods. In the rest of the FHIR Search Framework this functionality is only possible using prefixes on date search parameters. However, only special cases of overlapping can be matched. A generalization cannot be achieved.

Lastly, it is to be noticed that instead of making use of composite search parameters, a new idiosyncratic way of searching on a pair of values is introduced:

"<paramName>[<filter>].<paramPath> SP <compareOp> SP <compValue>".

Hereby, a filter can be used to execute a pre-selection of values. Only these value are still to be considered for matches of the search for <paramPath>.

Valid examples for _filter search requests can be found in Listing 23.

3.5.4. Text Search

While trying to achieve a fair balance between flexibility and having a sufficient degree of structuredness, FHIR enables interoperability in a systematical way through resources. Although a large part of the specification is optimized towards capturing information in specialized elements (preferably annotated with codes), FHIR consistently provides free-form text elements. In resources where it is not clear how to codify the involved concepts, respectively, if it is unknown whether or not it is even necessary or preferable to enforce such a feature, text can be expressed as annotations or strings. The resources ClinicalImpression, Condition or CarePlan exemplify this behaviour. All of the mentioned resources may contain a note which can include any additional information not representable through the other resource elements.

In general, indications have been presented in literature that improvements to health care exist if systems offer a high level of expressivity through having the option of recording clinical impressions as natural text [104]. When effectively integrated into the clinical workflow, clinical notes contribute to the establishment of continuity of care, as all involved stakeholders can access relevant cross-sectoral information about a patient. All in all, it facilitates the prevention of severe clinical incidents, e.g., adverse reactions [105].

Besides these advantages with regard to its flexibility, free-form text poses challenges in the context of searching and automated processing of the captured information. In order to not only search based on keywords but to allow systems to proactively understand what concepts are requested, advanced text mining is required.

Knowledge from free-form text can be derived after it been preprocessed by extracting concepts of interest (e.g., related names of drugs, or generalizations of accident reasons), combine them with machine-processable codes, and moreover identify the timeline of mentioned events within the text [106]. Without such an abstract view of the text, meaningful relationships cannot be created [107]. Despite recent advances in this field [80], there are still issues, mainly resulting from data quality [108], to be overcome.

Nevertheless, if a sufficient index of the content inside FHIR resources can be constructed, the FHIR Search Framework offers functionality of searching through it using the _text and _content search parameter. This parameter grants clients access to full-text search using keywords in combination with logical operators. Either the text in the narrative portion of a resource is used or the whole content of the resource is being treated as text.

3.5.5. Advanced Search

Due to the nature of an international standard, it is a challenge for the FHIR specification to foresee needs and requirements. In order to not limit the applicability or add unnecessary constraints, flexible and customizable solutions are needed. This statement equally applies to the topic of searching. As a whole, FHIR describes itself as "[...] a set of capabilities [for the] use across the healthcare process, in all jurisdictions, and in lots of different context[s]" [109]. Therefore, it cannot be optimized for specific use cases but may need to define only the minimum level of interoperability as a "framework".

In anticipation of this challenge, the FHIR Search Framework offered, starting with the first official version [110], a fallback option for searching. Using _query servers and clients have the choice to use custom named queries, which are essentially a custom operation, as described in section 2.2.2. Arbitrary search requests with freely definable content[13] may be defined and exchanged. The exact behaviour of a server implementation of a custom named query would need to be documented externally, for example using an Implementation Guide, to establish a cohesive semantical meaning of the query parameters across implementations.

3.6. FHIR Search Components Model

Most details and features discussed in this chapter are optional for implementation if conformity to the FHIR Search Framework is claimed. All parts can be combined up to the degree of necessity to resolve the information needs of users. In reference to the first and second research question of this thesis, the question arises if and how such an information need can be gathered and described independently of the syntax provided by the standard. A model fulfilling this purpose would enable a vocabulary to describe current pain points, i.e., illustrate which features are not optimally integrated in the schema defined by FHIR. Additionally, the search requests could be discussed on a meta-level. It would help to extend the possibilities for inspecting a search request, such that it could help users to improve the quality of their result sets by having a clearer overview of the implications of their requests.

[13] Search parameters and values do not necessarily need to refer to an element of a FHIR resource.

For this purpose, an abstract model to capture and describe the search requests can be build with the components described below. The feasibility of this approach is tested in Appendix E. Exemplary search requests, using possible features of FHIR Search Framework are modeled via the components described below.

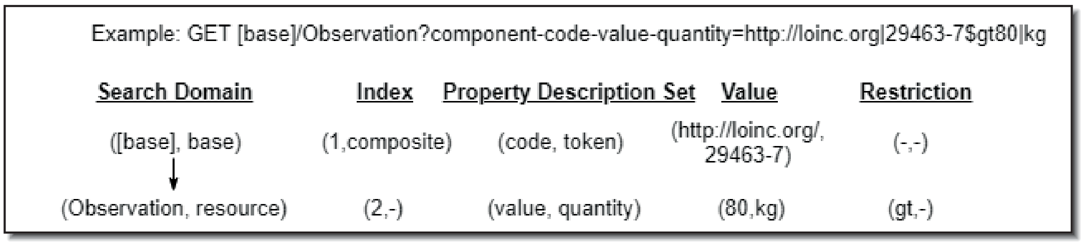
- **Search Domain**
  A search domain describes the relationship between each of the used search contexts. It indicates which set of resources is ultimately the target of a search request. A search domain can be used to model a target hierarchy in the following cases using an ordered pair with the following values:
  - Compartment searches: (ID of referenced resource, “compartment”)
  - Requests with resource-specific search parameters: (Resource type, “resource”)
  - Usage of _type: (Resource types, “_type”)
  - System-wide searches: (Base URL, “base”)
  - Chaining: (Resource type which is actually being searched, “chaining”)
  - Reverse Chaining: (Resource type which includes the common reference, “reverse chaining”)

  The hierarchy is explicitly indicated using an arrow notation. Please note, complex combinations are possible when modelling a search domain. For example, _type may be used with chaining and reverse chaining, due to the fact that cross-resource search parameters of type reference exist, and _has is a search parameter shared between all resources. Such an option must be taken into account when designing the model.

  It is, however, unclear whether or not chaining and reverse chaining can be combined. This issue is further discussed as a potential limitation of the FHIR Search Framework in section 4.6.
- **Index**
  An index describes the order of the given search parameters. Moreover, it helps to separate them based on the indented combination. Chaining, reverse chaining and logical ANDs can be expressed. The first item of the “Index” tuple is an ascending ID. The second item represents the used operator, that combines the current search parameter with the next one. It may consist of the following values:
  - A logical operator (AND / OR)
  - A navigational operator (chaining)
  - “composite”, indicating that the search parameter consists of two components
  - “-”, indicating no value
- **Property Description Set**
  A property description set is a 2-tuple consisting of the name of the search parameter and its type. Based on this categorization, a user can directly obtain a structured overview of the used search parameters, which can be of value if complex search requests with reverse chaining or composite parameters need to be created or comprehended.

- **Value**
  This component represents a value of a single search parameter including any additional sub-parts of the value (e.g., the system and code of a quantity).
- **Restriction**
  Restrictions can be used to capture either a modifier or a prefix of a search parameter value. Both restrictions cannot be applied at the same time for a single search parameter.

As a result, a summary, respectively, an overview of a FHIR search request can be gained, as depicted in Figure 18.



Example: GET [base]/Observation?component-code-value-quantity=http://loinc.org|29463-7$gt80|kg

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,composite)	(code, token)	(http://loinc.org/, 29463-7)	(-,-)
↓				
(Observation, resource)	(2,-)	(value, quantity)	(80,kg)	(gt,-)

Figure 18: FHIR Components Model Example

The relevancy for this model is given in the context of this thesis, as it may help a user to understand the extend of a single search request, how each parts of a search request is connected, and learn how to apply the individual components of the FHIR Search Framework (Modifiers, Chaining, etc.).

4. Current Limitations FHIR Search Framework

In summary, as presented in the previous chapter, the FHIR Search Framework provides an abstraction level above the resources elements. Through its flexible approach for selecting targets of search parameters, stability for the search interface can be established. Altogether, it helps to focus on searching for resources based on elements which are needed in most common use cases, and therefore it assists in coping with the variety of resources elements. Additional to these features, custom search parameters, named queries and _filter expressions contribute to a powerful extendability.

Based on the previous description, this chapter seeks to answer the first research question posed in this thesis: "Is the current FHIR Search Framework well-suited to solve tasks and challenges occurring within its current scope?". To enable a comprehensive and objective answer to such a question, it needs to be established which criteria should be fulfilled in order to categorize the evaluation target as "well-suited".

The current thesis is based on the hypothesis, that searching in FHIR may be labeled "well-suited" if a sufficient quality can be achieved when trying to explore resources. This includes finding specific resources or getting an overview over resources of a specific type. The exact sub-measures, which ultimately define how the quality is determined, are described detailedly in the next section.

Please note, that this hypothesis is not restricted to a specific group of stakeholders. Rather, quality should exist for all stakeholders of the Search API, which also includes users like patients. A standard should achieve an added value for all involved parties. Subsequently, quality is defined in accordance with the ISO 9000 standard as the "degree to which a set of inherent characteristics of an entity fulfills need[s] or expectation[s] that [are] stated, generally implied or obligatory" [111].

Building upon the idea of using quality as the main theme of this discussion, it is argued that quality metrics are most often directly applied to software, products or systems[14] and not standards. Still, the general nature of the definition helps to provide a ground for quantifying the effectiveness and efficiency of the FHIR Search Framework.

Similar to the ISO standards related to (software) quality, it seems reasonable to define a "Quality in Use" and a model for inherent quality aspects. The resulting model characteristics are discussed throughout this section. It starts with an analysis of factors which can significantly influence a user's view of quality when using the FHIR Search Framework. Lastly, the current chapter concludes by providing characteristics for measuring the intrinsic quality, i.e., the quality characteristics similar to the product quality model of ISO 25010.

[14] See discussion of ISO/IEC 25010 in section 2.1. Please note that ISO 9000 is related to ISO 25010, as the latter supersedes ISO/IEC 9126, which is part of the ISO 9000 standards family for quality management systems.

4.1. Quality In Use - FHIR Search Framework

"Quality in use" can be defined in the context of this thesis by combining the definition for quality with the definition for searching as given in section 2.4:

> Degree to which the overall group of FHIR standard stakeholders[15] can satisfy their information needs using the currently defined search functionalities.

The inclusion of the term "information need", in contrast to user wants, user needs and other implicit requirements referenced by the ISO standards, has a major effect on the measurement of the quality in the context of searching. Consequently, it is necessary to discuss the implications of an "information need" to identify core aspects which it contributes to the measurement of the frameworks' practical applicability, i.e., to the measurement of the user's success.

Commonly, an information need is defined comparably to the following description:

> "An information need is a recognition that your knowledge is inadequate to satisfy a goal that you have." [113, p.5]

The current thesis classifies the term "information need" as fitted to serve partially as the foundation of the established definition of quality. This assertion is based on the recognition that users in general, if not given a concrete requirement which must be fulfilled, are first and foremost driven by their motivation resulting from this need. Despite concentrating on direct users, it is clarified that such a focus should not result in a conflict with the fundamental design principles of FHIR, namely that the specification is written first and foremost for developers. This group is not excluded from contributing to the assessment of the FHIR Search Framework. However, their view on a quality in use does not play an immediate role. In contrast, their impact on non-functional aspects should be weighted more strongly.

Illustrating this point further, a concrete example of an information need with regard to health care may be a medical researcher looking to retrieve data which adheres to a precise description of some phenotype, e.g., an expression of certain patient-specific characteristics. In this example, the researcher is expected to act based on the need to retrieve the patient-related information. Hence, the information need is a predominant factor. A "sufficient quality" is necessary to describe the needed resources.

The difference to the ISO 25010 definition of quality in use results from having a different point of view on the problem space. Instead of focusing on the usage related aspects, the proposed definition above highlights that the stakeholder's main target is information-related and that the concrete implementation details (e.g., performance, efficiency of the search requests) of the executed search are only of subordinate interest. Nevertheless, to achieve adequate usability and user experience, i.e., support the user effectively, efficiently, and to general satisfaction, a "well-rounded" set of aspects should be assured by the involved systems and standards.

For searching and related operations, an information need is the starting point for any information-related inquiry. Associated processes are depicted in Figure 19.

[15] For a preliminary list of identified stakeholder from the perspective of HL7, please see [112]. Still, in other context, this list may be extended.

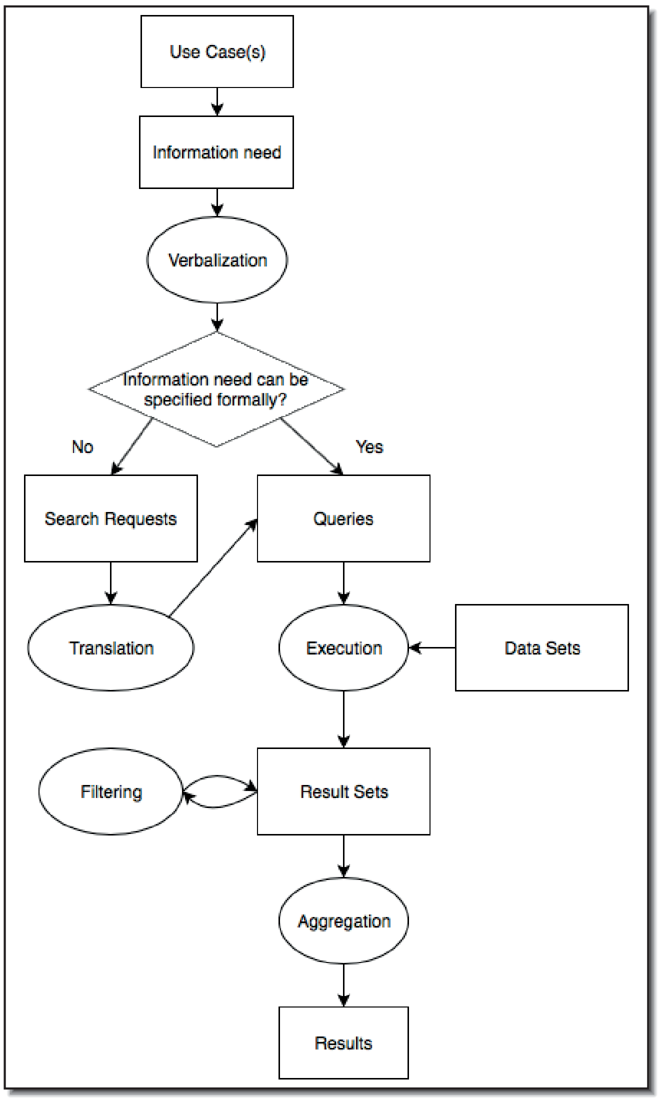


Figure 19: Information Need in the Context of Searching and Related Operations

It is to be noticed that ISO 25010 defines a series of characteristics around the topics of effectiveness, efficiency, satisfaction, safety, and usability for measuring quality in use. However, in the light of a standard, certain characteristics are not appropriate. Therefore, only a selection of these characteristics or their sub-characteristics are adopted:

- Effectiveness: The accuracy and completeness with which users can satisfy their information needs.
- Context coverage and Context completeness: Degree to which an information need can be satisfied in each of the specified contexts of use, as well as in contexts beyond those initially explicitly identified.
- Safety: Degree to which a search framework supports the user to mitigate errors and the accidental inclusion of unintended search matches.

Excluded are the following characteristics:

- Efficiency: The efficiency of searching depends on its concrete implementation. However, such an assessment is highly context-dependent, as the scale and target objective for efficiency may vary strongly per use case. Efficiency of the standard itself is to be interpreted as the efficiency in representing a search request. The limiting factors in optimizing this criteria are interoperability and unambiguous expressiveness. With regard to the overall aim of the standard, these goals need to be weighted more strongly. If efficiency is a limiting factor in the final implementation, it should be treated as a satisfaction challenge (see next point).
- Satisfaction: Due to the nature of a standard, individual perceptions of the user cannot be taken into account significantly. If the standard is deemed as inadequate, change requests must be submitted for improvement. The standard, which can be regarded as a trade-off between the needs, wants and requirements of a large group of stakeholders, should not be evaluated based on such a subjective characteristic.

As can be inferred from the definitions above, the measurement of completeness takes an important role. For this reason, this thesis aims at contributing to an analysis of areas which may contain optimization potential.

The complete FHIR Search Framework, as depicted in Figure 14, can be grouped based on similarities in functionality. For each of the resulting categories, individual criteria for completeness are established. If additional functionality would theoretically be possible in one of the defined areas, an exemplary description of how this feature could be included in a search request, as well as an assessment of its effects is given. However, it is to be noticed that this analysis is solely meant to start a discussion about the limits of the FHIR Search Framework. In a consecutive debate, it is to be evaluated if these proposed features are feasible and desirable. Therefore, they are only meant as a "food for thoughts", in order to determine to which degree the RESTful API is to be extended. The standard should not be broadened for the sake of completeness, the main purpose of enabling a framework for effectively and efficiently dealing with health care data should be the primary focus. All additions should alleviate pain points in concrete use cases.

Please note, in the next future edition of the FHIR Standard (FHIR Release 4), changes have already been made to the Search API. Some of these changes also reduce present limitations. However, to avoid redundancy, these changes and solved limitations, even if falling into the current area of interest, are not listed below. Unfortunately, at the time of writing the current thesis, no reliable changelog or comparison can be included due to the fact that the next version of FHIR is still a moving target.

An information need can be expressed and resolved in FHIR through the following categories of features of the "classical" search requests and by using mechanisms like _filter or _query:

- **Establishing and accessing a search context**
  All search requests are meant to be evaluated within a particular scope. Differentiating between these contexts, based on the different search interactions, and defining search parameters for them are essential tasks for the FHIR Search Framework.

- **Explore resources**
  Search Parameters excluding corresponding modifiers and prefixes are defined to retrieve the content of a resource.
- **Adjust search behaviour**
  The search parameter types and all defined modifiers and prefixes adjust the behaviour of the search parameters.
- **Advanced queries**
  _text, _query, _filter and composite search parameters are features based on which complex search requests can be executed.
- **Advanced traversal of resources**
  Chaining and Reverse chaining can be used to traverse resources based on common links.

In the next section, the list formulated above serves as the basis for evaluating the completeness of the FHIR Search Framework, by assessing the quality in use of each category.

4.2. Evaluation: Establishing and Accessing a Search Context

A two-fold approach has to be taken when analyzing the completeness of scope-related features. Completeness can be defined in this category through:

- Completeness resulting from being able to differentiate between different search contexts.
- Completeness with regard to which search parameters are applicable in the different search contexts.

Taking all features into account, there are only a limited number of restrictions in terms of functionality for the different search contexts:

- It is currently ambiguously defined which search parameters can be used on a system level with _filter. Therefore, inconsistencies can occur in search behaviour resulting from an invalid interpretation of filters if not fully recognized in this search context. Undefined behaviour can result from the fact that no explicit guidance is given by the standard in terms of how to handle this parameter on a system level.

  As a general matter, _filter is defined as a parameter for all resources. Consequently, it is applicable on a system level. Additionally, there are no defined restrictions for limiting which search parameters represent a valid "path" in a filter expression. This issue materializes itself in the following search request structure: "[base]/?_filter=path operation value&parameter". As "path" is conceptually different from "parameter" it does not inherit the restriction to only be substitutable with search parameters which are common to all resources or have a shared "base". Due to the absence of such a rule, a mix and match of search parameters from across resources is possible. Having this loose definition of a path will result in semantical ambiguities as it is not specified how to evaluate the paths. For example, when executing a search on Organization and Patient with the path "name", it is unclear whether or not the equivalent search parameter in both resources should be evaluated.

More complex challenges arise when dealing with paths that refer to overloaded search parameters like patient, which can be used as clinical-patient and Account-patient. Both search parameters refer to different targets. Additional type information must be added for these requests on a system level, such that it is uniquely identifiable to which resource a search parameter belongs and therefore which resource elements are targeted. Another option would be to restrict paths in the same way as all other search parameters on a system level, respectively to define evaluation rules for how to interpret ambiguous search parameters.

Furthermore, the specification does not fully utilize its expressive power when it comes to defining _filter and other search parameters. To foster interoperability and to dynamically determine system behaviour, information about the applicability in different search contexts should be included in each of the machine-readable SearchParameter resource. To borrow ideas from the OperationDefinitions resource, it would be possible to add a boolean to define the applicability for each respective search context. Please note that the standard should evaluate whether or not a compartment is different from the system level, as no boolean exists for the purpose of differentiation in an OperationDefinition.

- From the previously listed restriction, it can be derived, that it is currently not possible to completely describe the execution context of a named query. The compartment context cannot be expressed. It is to be discussed to which extent the compartment level is different to the system level and if for this purpose an additional boolean is required in the OperationDefinition resource.
- Despite the listed justification in [114], it has to be noticed that it is not possible to define custom compartments. Having the above-described restriction effects searches which would be based on shared elements which are not references and therefore could not be used to form a compartment. For example, search for a specific SNOMED CT concept in all resources containing a “code” element (besides Medication resources).

  Only inefficient substitutions exist in cases where it would be necessary to execute such a search request. The only formally correct option would be to list all resources types having a code parameter in _type. Such a missing capability is noteworthy, as the Search Parameter Registry, practically, even highlights cross-resource search parameters. These parameters can be achieved as it is allowed that a single search parameter defines that it applies to an arbitrary selection of resources. The “base” element (Cardinality: 1..*) of a SearchParameter can be used for this purpose. Nevertheless, they can only be used in combination with an enumeration of all needed resource types provided with _type.
- When executing a system level search in combination with _type parameter, it is not possible to provide semantical “clues” for the system to resolve ambiguities in terms of which search parameter is referenced. If existing, it would be possible to soften the restriction that only common search parameters could be used. The semantics of this search request should express a search with an implicitly applied _type parameter. It would harmonize the syntax for accessing cross-resource search parameters, as it could be used in _filter, when restricting compartments to a selection of types, and in any other use case where currently _type could be used.

An idea for implementing such behaviour could be providing resource names as prefixes in front of the search parameter. A joint result set for all types is to be returned, similar to the behaviour when executing requests with multiple reverse chaining parts.

4.3. Evaluation: Explore Resources

Completeness for exploring resources, i.e., searching for specific content in one resource, is given in the sense that is possible to search for the most commonly used elements via the officially defined search parameters and add extensions if needed via custom parameters and names queries. System interoperability can be checked by parsing a system's CapabilityStatement and dynamically determine if a search parameter is supported.

Interoperability may only be limited if these checks are not properly implemented by clients. In this regard, complexity is outsourced to implementers to provide flexibility. As a result, critical situations may result where a certain sever does define a search parameter differently from the standard. Please note that overriding a search parameter is possible as it is not directly forbidden. The standard only recommends starting a custom search parameter with a "-".

Completeness of enabling the search on all FHIR-defined datatypes is fully provided and retraceable through a cross-mapping of the FHIR data types used in the resources and the search parameter types. Such a mapping is provided by the specification since FHIR DSTU 2.

4.4. Evaluation: Adjust Search Behaviour

FHIR provides a range of modifiers and prefixes to customize the behaviour of a search parameter. Both of these syntactic elements contribute directly to the effectiveness of the FHIR Search Framework as it increases the accuracy with which search requests can be expressed. Completeness for prefixes is given by providing all prefixes necessary to describe any point in a specific range. This holds true for any numerical search parameter type. Modifiers should be subject to a more nuanced discussion. This concept combines two purposes into one element of search requests:

- Search on a specific sub-element of a search parameter referring to a complex FHIR resource element. For example, :text is a specialization of token search parameters.
- Other modifiers are shorthand methods for processing and filtering the final result set. The defined modifiers therefore only provide a syntactical element for efficiently representing these operations.

Due to the powerfulness of modifiers, any potential extension is to be critically evaluated in the light of FHIR's core principals. The fact of not having a general composition framework for creating new modifiers is to be compensated by outsourcing the complexity of filtering to the client or providing in-line filter statements. Another valid substitution for the first use case of a modifier as listed above is to define a custom search parameter, which can point to any desired element with a suitable FHIRPath statement.

A parity between _filter and the regular search interactions may be pursued on the level of modifiers by introducing the :not modifier on other types than tokens. Hereby, simple logical evaluations may be possible in an interoperable way even if “full-featured” logical expressions are not implemented.

Furthermore, it is suggested that the applicability of aliases of CodeSystems should not be limited to _filter, due to usability gains resulting from improved manual creation of search requests. To provide interoperability for this search detail, a new resource element “alias” may be added to the CodeSystem resource in addition to providing a fixed enumeration of external and well-known CodeSystems like LOINC and SNOMED CT.

4.5. Evaluation: Advanced Queries

The following discussion about advanced queries is only meant in reference to currently defined functionality. Completeness is to be evaluated within this boundary. As stated previously in this thesis, the main purpose of the FHIR Search Framework is to enable the retrieval of resources based on their structural properties. It is not meant to serve as the basis for executing requests for which it would be necessary to reason about the content of the resources.

By this is meant any “meta” query with abstract expressions are needed, which would need to be transformed to parts of a concrete FHIR resource element. For example, “Search for all Patients who are older than 18 years.” cannot directly be expressed as it would need to be transformed to “?birthdate=lt2000”. Still, it is noteworthy that a broad set of these queries could be translated to a FHIR Search requests, as even complex inquiries are representable. The concrete extend is evaluated in the next section.

For all other non-defined purposes, _query is left as a fallback option for implementers with more specific and custom needs. With regard to evaluating functional completeness, especially, the standards decision to not predefine any syntax for it is to be respected.

Essentially, similar design decisions are also made for other search parameters for advanced queries. Due to a current lack of feedback from implementers, the scope and features of _text are limited. Consequentially, simple text search operations like excluding a certain term, searching with wildcards or sub-searches for terms only in specific resources are not specified.

Other advanced search parameters exhibit critical challenges in terms of completeness as well. In the case of _filter, these issues arise from its formal grammar and definition:

- **_filter does not allow search parameter starting with “_”**
  The formal grammar specifies that a paramName in a path must start with an alphabetic character. From such a definition follows that search parameters starting with a “_” are not valid as paths. This implies that most of the search parameters which are designed to be valid for all resources cannot be used in a filter context.
- **_filter is not formally defined through a SearchParameter resource**
  The official Search Parameter Registry of the FHIR specification does not list _filter as a valid search parameter. Additionally, there exist no officially published SearchParameter resource. Therefore, no computable verifiability of _filter support can be gained.

- **_filter does not allow reverse chaining**
  When using _filter, paths can be chained through a dot notation as in the regular search requests. Still, reverse chaining is not defined as part of a filter due to restricted use of “_”-characters in search parameters. The only reference operation which is provided (re), operates on a given reference value. Therefore, no feasible replacement option can be built. As a result, an integral part of the functionality for traversing FHIR resources is not accessible.

Unrelated to the FHIR specification but linked to _filter, it is to be reported that implementation issues exist in relation to this search parameter. To the best knowledge of the current author, _filter is only implemented in the Delphi Reference Library [115].

In the current version at the time of writing (Version 1.0.238 built 2018-06-04), _filter does only allow uncommon syntactical constructs when grouping logical terms with parentheses. In a variety of programming and query languages the precedence of logical operations can be expressed explicitly by grouping terms with parentheses. Functionality wise, _filter does not contain any restrictions in this regard.

Generally, it is common to write logical expressions in a similar form to:

$$((\text{A or B}) \text{ and C}) \text{ OR D}$$

The presented logical grouping should be representable with _filter, whereby the variables A, B, C, and D would be substituted with a search parameter expression in the form of “paramPath SP compareOp SP compValue”. It can be shown through the following derivation steps based on the formal grammar of _filter that the above-mentioned logical expression is valid:

$$\begin{aligned}
\text{filter} &\Rightarrow \text{logExp} \\
&\Rightarrow \text{filter or filter} \\
&\Rightarrow \text{(filter) or filter} \\
&\Rightarrow \text{(logExp) or paramExp} \\
&\Rightarrow \text{(filter and filter) or paramExp} \\
&\Rightarrow \text{((filter) and filter) or paramExp} \\
&\Rightarrow \text{((logExp) and paramExp) or paramExp} \\
&\Rightarrow \text{((filter or filter) and paramExp) or paramExp} \\
&\Rightarrow \text{((paramExp or paramExp) and paramExp) or paramExp}
\end{aligned}$$

Despite being able to generate the expression, the current _filter implementation raises exceptions when the groupings are used differently as presented in Appendix D. Being only able to use parenthesizes after the first logical operation may seem unconventional.

4.6. Evaluation: Advanced Traversal of Resources

The following section asses if challenges can occur resulting from potential limitations when traversing FHIR resources through reference search parameters. As a prerequisite, a graph model is defined to represent the interconnectedness of FHIR resources. Based hereupon, well-known graph query expressiveness models are applied with the goal of determining search requests which cannot be expressed with the current search syntax.

4.6.1. Chained Search Graph Model

All necessary information to resolve the target of a chaining or reverse chaining search request can be conveyed by a labeled multigraph G where each vertex v_i represents an individual instance of a resource. Instances are connected by one or more directed edges. An edge is to be created if the resource instance of the source vertex contains a reference to the target's instance. Edges must contain a valid reference search parameter name as their label which corresponds to their represented reference.

The actual value of the reference is not of interest for resolving a chaining or reverse chaining request, as the interconnectedness of the resources is modeled in the search request by the search parameter name and not by value.

References in an instance which do not have an existing counterpart, i.e., where the target instance does not exist, must be connected to a "ghost" node, which indicates the absence of the referenced instance. This addition is necessary as search requests can operate on the chain of references itself and do not always involve the data of the target instance (e.g., it is possible to check for the existence of a chain with the :missing modifier).

In order to not unnecessarily complicate the underlying model, the actual data of the resource instances is not considered in this graph model. From the viewpoint of a chaining search, it is sufficient to know the id of the references resource, which can be modeled as a label of the node. However, if necessary the graph model could be extended to represent complex nodes with data inside or let each node contain an additional link to a separate node comprising the data.

Formally, a chaining (chaining and reverse chaining) search graph can be represented using a standard definition of a multigraph G [116]:

$$G = (\Sigma_V, \Sigma_E, V, E, s, t, l_V, l_E)$$

V can be defined as a non-empty set of vertices: $\{v_1, ..., v_n\}$, $n \geq 1$. Each vertex is connected by one or more labeled and directed edges $(v_i, v_j)_{v_a} \in E$ with v_a as the edge label. v_i may equal v_j (loops need to be allowed in the model, as demonstrated later on). Σ_V is the finite alphabet of all IDs used by the connected FHIR resources. Σ_E is the finite alphabet of all used reference search parameter names. Please note that Σ_V and Σ_E are disjoint sets. s and t are functions indicating the source and the target of an edge: $s : E \to V$ and $t : E \to V$. $l_V : V \to \Sigma_V$ and $l_E : E \to \Sigma_E$ are functions assigning each edge and vertex its label v_a.

An example of a graph as defined above is depicted in Figure 20. Please note, that loops and cycles can exist in a chaining search graph as shown in Figure 21, even if the associated semantical meaning may not be consistent.

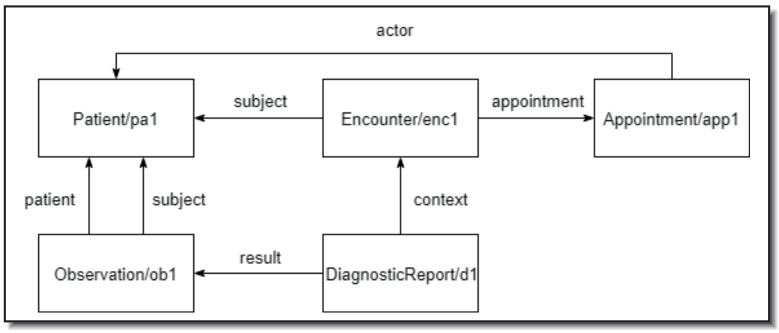


Figure 20: Example Chained Search Graph

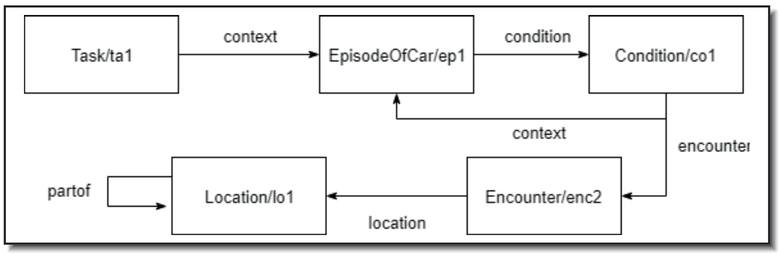


Figure 21: Example Chained Search Graph with Loops

A chaining search requests (e.g., [base]/?DiagnosticReport.result.patient.name=Jane) can be resolved by a) checking if a corresponding path exist in G by traversing the indicated path and b) checking if the instances of any matching path contain the name Jane in the corresponding resource element. The instances can be resolved via the IDs given in the vertex labels.

Reverse chaining search requests can be resolved similarly. More details for each type of a chaining search can be found in the subsections below.

4.6.2. Regular Path Queries (RPQ)

On an abstract level, chaining can be interpreted as a query on a graph for determining reachability. Basically, a FHIR server is expected to answer the question if there are any paths connecting vertex x and y. If yes, the instances of the source vertices of any matching paths are to be returned. A path is to be defined as a finite alternating sequence of vertices and labels.

As described in detail by [117], there exist different classes describing the expressiveness of the queries which can be used for path matching. Based on the arguments presented in [118], the following hierarchy holds true for the query expressions, due to the fact that each category extends the possibilities for queries of the respective previous class:

$$\begin{gathered}\text{Regular Path Queries} \subset \text{Two-way Regular Path Queries}\\ \subset \text{Conjunctive Regular Path Queries}\end{gathered}$$

In the following sections, the individual capabilities of the different forms of graph queries are presented. Subsequently, it is evaluated which of these features can be mapped to an expression of the FHIR Search Framework. Any occurring gaps are acknowledged and assessed with regard to their individual impact towards inexpressible chaining search requests.

Basic navigational questions can be answered in a graph through Regular Path Queries (RPQs). RPQs are expressions in the form of (x,R,y) where R is a regular expression, indicating a query for the existence of a path between node x and y.

R must consist of a combination of the following operators / expressions, as its expressiveness is restricted to the class of Regular Languages:

- A single element s $\in \Sigma_E$
- A concatenation of R $\cdot$ R
- An operator for a selection R|R
- The Kleene star operator R* and its variations R+ (R $\cdot$ R*) and R? (R|ϵ)
- Parenthesis symbols (R) for grouping purposes

A RPQ matches if there is a path p in G containing a path label $\lambda(p) \in L(R)$. $\lambda(p)$ is a string s_0 ... $s_{m-1} \in \Sigma_E^*$ where m is equal to the length of p.

Projecting these operations onto the FHIR Search Framework, it is recognizable that chaining equals the concatenation operation listed above. The appliance of a check for a single element equals a search expression using the :missing modifier or chaining over a single reference search parameter.

The evaluation of the graph query (x,R,y) is made easier by the fact that the types for x and y are known when resolving a chaining search request. More importantly, the number of nodes, which need to be considered as a potential match of a sub-expressions of R (e.g., a single chaining step) can be reduced by having relevant type information.

Every other operator listed above, which is theoretically usable in a RPQ, is not accessible for search requests using the “regular” FHIR Search API. For more advanced RPQs, a _filter expression must be used. As a result, users are mostly confronted with the limitation of not being able to apply a selection and logical grouping. With _filter, these restrictions are lifted to some extent. Logical groupings are provided within the _filter syntax, and selections (disjunctions) can be written using logical OR operators. However, it is not possible to use the recursive repetition of a reference in an arbitrary way; thus R* cannot be described with the currently defined syntax elements. Nevertheless, this resections should not be a rigid limitation in practice as recursive references (loops) are currently not widely spread within FHIR resources. R? can only be used by searching on the whole chain including the optional part and on a disjunction with the same chain without the optional part.

4.6.3. Two-way Regular Path Queries (2RPQ)

2RPQs defines the class of all path queries which include the traversal of edges regardless of their direction. In order to achieve this capability, R is extended to allow inverse operations. Formally, the alphabet Σ_E is broadened to include a symbol s^- for every s. This alphabet is labeled $\Sigma_E^{\pm}$. s^- is matched as part of a path if there exists a corresponding incoming edge from the target to the source vertex.

For example in Figure 20, the following path query returns true, as a path exist with incoming edges from DiagnosticReport/d1 to Patient/pa1:

$$(\text{Patient/pa1, patient}^- \cdot \text{result}^-\text{, DiagnosticReport/d1})$$

The inverse operation itself can be mapped to the _has search parameter. Nonetheless, it is not possible to represent the whole 2RPQ class. Issues arise from missing support for reverse chaining in _filter. Equivalent to RPQs, there is no way to express a disjunction of two sub-paths. Additionally, all other limitations for RPQs are still valid for 2RPQs.

Furthermore, it has to be noticed that _has is not officially defined in the FHIR Search Parameter registry [119]. For that reason, its type cannot be determined. Moreover, FHIR specifies that chaining can only occur on reference search parameters. If not defined as the aforementioned type, _has cannot (technically) be combined with a chaining search request. The direction of edge traversal cannot be switched within a search request.

4.6.4. Conjunctive (Two-way) Regular Path Queries (C(2)RPQ)

Conjunctive (Two-way) Regular Path Queries are, as the name suggests, based on an AND combination of multiple RPQs / 2RPQs:

$$anz(\overline{z}) \leftarrow \bigwedge_{1\leq i\leq n} (x_i, R_i, y_i) \text{ with n} > 0.$$

Each (x_i, R_i, y_i) is a (2)RPQ where x_i, as well as, y_i are free node variables. A query is answered by a returning $\overline{z}$. $\overline{z}$ is a arbitrarily selected tuple of variables from $\overline{x} = (x_1, ..., x_m)$ $\cup\, \overline{y} = (y_1, ..., y_m)$. A C(2)RPQ is evaluated by checking if for each $1 \leq i \leq n$, it is possible to find a mapping h (maps free node variables to concrete nodes), such that the mapping is the answer to (x_i, R_i, y_i). A formal definition of answering a query in the described form is given in [120].

For example, given the graph in Figure 20 and the following query Q:

$$\text{anz(x) <- (x, subject}^-\text{.result}^-\text{, y)} \wedge \text{(y, context.appointment.actor, x),}$$

the answer would consist of the value “Patient/pa1” as a valid value for the node variable x. The mapping h would exist, because x can be bound to the Patient node “pa/1”, y to DiagnosticReport “d/1”, and y is connected through the given chain to x, where x has the same value as previously assigned. Please note, that it would theoretically be possible to select anz(x,y) as the answer, due to the fact that any desired subset of the used node variables can be chosen.

Given the definition of C(2)RPQs, two components need to be mapped to the FHIR Search Framework for comparing if approximately similar concepts exist: a) how to select $\overline{z}$ and b) how multiple (2)RPQs can be combined via conjunctions. Resulting challenges for representing such concepts are discussed further below.

For the purpose of examining the general feasibility, a differentiation has to be made between _filter and the regular Search API. Yet, in any case, before trying to analyze how to implement the requirements imposed by C(2)RPQs, it may be helpful to understand the functional gain resulting from this expressiveness category.

On a fundamental level, C(2)RPQs add the ability to join subgraphs based on simple equality comparisons. In all previous expressiveness classes, it was only possible to operate on a single (implicit) resource type. Now, a selection of resources with varying types can be queried. In doing so, a collection of (2)RPQs can be executed against the different types simultaneously. The equality comparisons come into effect by additionally enabling to enforce that node variables among different (2)RPQs share the same value, effectively indicating a join of two subgraphs using the selected node.

When discussing the implementability of C(2)RPQs, the issue of selecting $\overline{z}$ for returning the result set is considered first, since it represents an essential feature of this expressiveness class. Given the summary in section 3.1, the _type search parameter, in combination with _include and _revinclude[16], could (theoretically) be used to limit the response of a search request to a selection of resource types, essentially mirroring $\overline{z}$.

However, since the semantics of these search parameters are designed to indicate an extension of a search scope (_type) or to include existing references without having the option to define restrictions upon these elements (include / revinclude), difficulties arise.

First and foremost, it is only possible to start a chaining search request with a search parameter which is defined on a resource type that is included in the enumeration of _type. Concludingly, with regard to C(2)RPQs, it can be stated that within the FHIR Search Framework it is not possible to form separate sets for the return variables and the free node variables used in the search request itself. In terms of a C(2)RPQs, $\overline{z}$ must contain all elements of $\overline{x}$. For example, when searching on "(a,R,b) $\wedge$ (c,R,d)", only (a,c) can be selected as the head of the query. For example, when searching with _filter and _type on Medication and Encounter using chained expressions, both resource types must be present for the search to be valid. Meaning, it could not selectively be chosen that only one resource type is to be returned. Please note, that the restriction of only being able to use cross-resource search parameters still applies, with the exception of using _filter.

From a practical point of view, these issues are less significant, as in many cases the exclusion of any $x_i \in \overline{x}$ hints at a semantically irrelevant Search Parameter, which can be left out of the query without consequences as it is redundant and does not contribute to a refined search.

[16] Up to this section, both _include and _revinclude have not been described yet. Both are so called search result parameters. They allow other resources to be included in the result set which can be specified through a "join" based on a reference search parameter.

This would only be not the case, if the set of resource IDs from a resolved (2)RPQ could be accessed and used in subsequent queries, i.e., the result set of two (2)RPQs could be used to form a join using a set of particular references. However, as shown below, such a capability does not directly exist.

Multiple (2)RPQs could be supplied in FHIR using a "regular" search request by providing chaining criteria combined with "&". Please note, that for reverse chaining, each chained parameter is processed individually, effectively representing an OR between each part. An AND combination cannot be used until _has support is added for _filter.

By contrast, _filter supports multiple RPQs by combining them using a logical AND operator. This feature lays the foundation for covering the current expressiveness class. (2)RPQs cannot be formulated as long as _has is rejected by the _filter grammar.

Considering the issue of supporting the selection of $\overline{z}$ further, _include and _revinclude could form the basis for including node variables from the right-hand side of each query in the search result. Having said this, momentarily, restrictions emerge when trying to specify such an inclusion.

It is especially noticeable that including references based on chaining expressions or including resources in the result set which are an intermediate result of a chaining expression is not specified. Instead, _include or _revinclude always follow their basic forms, hereby limiting arbitrary joins:

[base]/?<SearchCriteria>&_include=<ResourceType>:<reference>
[base]/?<SearchCriteria>&_revinclude=<ResourceType>:<reference>

Even if "<SearchCriteria>" represents a C(2)RPQ, the _include, or _revinclude, according to its original purpose, adds all resources to the result set which:

- Have an outgoing reference from the matched resources to another resource of type <ResourceType> through the link <reference> (include).

  Example: [base]/MedicationRequest?_include=Medication:medication
- Are referenced by resources of type <ResourceType> through <reference> (revinclude).

  Example: [base]/Patient?_revinclude=Provenance:target

In summary, the only C(2)RPQ that can be formed is in the form of:

$$\text{anz(a,c) <- (a, subject}^{-}\text{.result}^{-}\text{, b)} \wedge \text{(c, context, d)}$$

Even in this case, it is most likely that only _filter can be used to express such a search request. The regular Search API can only be used if all chained requests are performed on references shared between all mentioned resources.

All other combinations of selecting subsets, even with the support of _include and _revinclude, cannot be mapped to a FHIR search request:

$$\text{anz(c,b) <- (a, R, b)} \wedge \text{(c, R, d)}$$
$$\text{anz(b) <- (a, R, b)} \wedge \text{(c, R, d)}$$
$$\text{anz(b) <- (a, R, b)} \wedge \text{(b, R, c)}$$

Applying these results on concrete examples means the inexpressibility of the following pseudo search requests:

- [base]/?_type=Observation,DiagnosticReport&_filter=based-on:MedicationRequest .medication=<SearchExpression> and result.subject=<SearchExpression> &_include:recurse=Medication:medication
- [base]/?_type=Observation,Encounter&_filter=context.appointment=<Search Expression> and <PreviouslySelectedAppointment>.actor=<SearchExpression> &...

The first example listed above fails due to not being able to include Medication resources in the result set through the recursive include statement.

Only references which fulfill the following requirements can be added:

- The reference needs to be accessible through a reference search parameter
- The aforementioned search parameter is defined on one or more resources types listed in the enumeration of _type
- If the previous requirement is not applicable, the aforementioned search parameter must be defined on a resource type which can be reached recursively, starting from resource types listed in the enumeration of _type

In the presented example, MedicationRequest must be additionally included via its own _include statement, such that the whole search request becomes valid. This supplementary resource type ultimately contains the "medication" reference. For implementers, this results in the advantage of being able to selectively reduce the result set and narrowing down the possibilities of resource types that come into question for joining. This optimization alleviates the need of comparing the set of all resource types involved in any chaining expression with the base of the search parameter used with _include.

Please note, it is at the discretion of the server to define the maximum level of the nesting. Servers may even choose not to honor include parameters. All in all, the defined behaviour leaves the client with a multitude of resources in the result set, forcing a filtering on the client side. Such an approach becomes infeasible if a large number of unnecessary / unwanted resources are being included due many include statements or cross-resource references being matched unintentionally.

In addition to the described challenge, further use cases, like the second example portrayed above, would be solvable by defining that intermediate result sets, i.e., the result set which represents an answer to a single (2)RPQ, can be referenced by a unique identifier or some equivalent concept. This would result in the expressibility of the following search request or any derivation of it, as the identity of "b" can be guaranteed across the different (2)RPQs:

$$\text{anz(a,b) -> (a, context.appointment, b)} \wedge \text{(b, actor, c)}$$

The relevance of this issue results from the fact that the resource element at the end of the chain can contain additional restrictions, such as modifiers and prefixes. To reduce a duplication of the whole path, a way of referring back to a path needs to be established. Please note, that the last element of a chaining expression does not necessarily need to be of type "reference".

4.6.5. Data Path Queries

Similarly to the issue described in the last section, it would also theoretically be possible to refine a result set of a single (2)RPQ based on restrictions for the resources which contain the chaining references. Please note, this behaviour would represent an extension of the current approach. In the current version of FHIR, it is only defined that the end of the chain may be used to provide further constraints. All other parts of chaining expressions must be reference search parameters without modifiers (except :[type]). Therefore, every resource with the given reference search parameter is taken into consideration when evaluating a chaining request.

Given the feature of apply prior restrictions to a (2)RPQ, the chain is only to be resolved on references which link resources that can fulfill all criteria. For example, when trying to check the reachability of "context.appointment.actor", requirements for the involved Encounter and DiagnosticReport could be established, reducing the number of resources that are worth considering. Having this kind of capabilities was proposed in general for graphs by [121] as Data Path Queries. For this kind of query to work, the definition of a path in a graph must be extended to contain data values alongside the topology information which has been used for queries in the previous definitions.

For current implementations, only a "trick" involving custom search parameters can be used to mimic Data Path Queries. A specific set of references can be chosen for chaining by incorporating the selection criteria into the search parameters' FHIRPath expressions. Instead of just selecting the path, extra criteria can be given via a where-statement in the form of ResourceType.where(criteria).reference.

4.6.6. Overview Restrictions Graph Model

In summary, the following restrictions apply for chained search requests in general, regardless of which search parameters are available:

	Regular FHIR Search Framework
RPQ	- Selection (Disjunction): $R_1 \mid R_2$ - Recursion: R* - Derived variants of R* (R+ | R?)
2RPQ	No restrictions
C(2)RPQ	- Selection of return types - Mix and match of search parameters across different resource types - Joins using resources selected through a chaining expression
Data Path Queries	- Restriction for elements which are part of a chaining expression

Table 5: Evaluation Graph Expressiveness Regular FHIR Search Framework

	_filter
RPQ	- Recursion: R* - Derived variants of R* (R+) - Direct support for R?
2RPQ	- R^-
C(2)RPQ	- Selection of return types - Combination of 2RPQs - Joins using resources selected through a chaining expression
Data Path Queries	- Restriction for elements which are part of a chaining expression

Table 6: Evaluation Graph Expressiveness _filter

4.7. Evaluation: Non-functional Criteria

Independent from a quality in use, criteria can be established for measuring the inherent quality of the FHIR Search Framework. It can be evaluated on this basis to which degree users are supported in resolving their information needs through non-functional aspects of the standard. Overall success may be defined as being able[17] of achieving search requests which are relevant to the current search execution context, i.e., the context in which an information need exist for the user implementing or executing the search.

In the light of the previously presented arguments and with regard to the product quality model of ISO 25010, the issue of searching in FHIR can be considered as an usability issue. The user of the standard (regardless on which abstraction level, i.e., regardless if it is a developer or a end-user) is to be guided to achieve an understanding of the specified functionality. For example, when implementing the FHIR Search Framework in an PHR context, the same rules and combination possibilities for components of the API apply, compared to an implementation for retrieving FHIR resources for some more technical developer-related purposes. Therefore, the abstraction of the search is to be broken down into conceptually clear concepts for each type of user.

The overall usability could be measured through the following sub-measures as defined by ISO 25010. Slight alterations have been made to accommodate for small idiosyncrasies when applying the measures to a standard and not to a product or a system:

- **Appropriateness recognizability**
  Degree to which users can recognize that the standard defines appropriate functionality which can be implemented to resolve their information needs.
- **Learnability**
  Degree to which the standard can be used by implementers to achieve a learning experience resulting in an appropriate understanding for applying the standard such that information needs can be resolved with effectiveness and freedom from risk.

[17] In the sense of acquiring needed knowledge, and being supported by a system, respectively.

- **Operability**
  Degree to which the standard has attributes that make it easy to implement and apply practically.

Please note that satisfaction and efficiency are not included in the last enumeration, due to the same reasons as presented in section 4.1.

These listed measures form dependencies to provide a clear pathway to support users in achieving their goals. The resulting hierarchy of non-functional qualities is depicted in Figure 22. Initial ideas how to support these measures from the perspective of the standard are presented, too. Chapter 5 elaborates them in more detail.

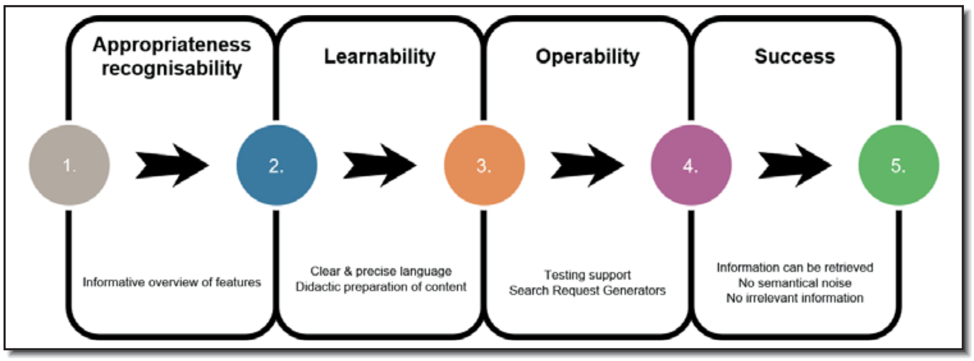

Appropriateness recognisability
Learnability
Operability
Success
1.
2.
3.
4.
5.
Informative overview of features
Clear & precise language
Didactic preparation of content
Testing support
Search Request Generators
Information can be retrieved
No semantical noise
No irrelevant information

Figure 22: Usability Quality Measures

On the other hand, other non-functional aspects which cannot be directly influenced by FHIR play a central role as well and should not be neglected. For example, it is essential for achieving quality results, that semantic noise (ambiguities in the search requests) is reduced. Such a tasks is independent of the used search syntax and how it is being presented by the standard. It does not fall into the realm of the standard, yet it is a non-functional criteria for quality.

5. Information Need and Usability

Depending on the involved stakeholders, the task of searching in FHIR can also be regarded as an usability, respectively, interaction design challenge. Originating from the goal of increasing patient satisfaction and achieving high-quality health care, it is necessary to not only integrate all crucial data into the health care process, which revolves around the patient, but also to offer possibilities for engaging with such data.

Metaphorically speaking, each stakeholder, may it be a patient, physician, or data scientist, initiates a “conversation” with its corresponding interaction medium (PHR, EHR, databases). The course of this conversation needs to be adjusted by the system towards achieving the users personal goals, which is most likely resolving information needs.

For this to work, it is indispensable that the need and wants of all involved parties are considered. Thereby, the way of interaction can be coordinated, enabling a cohesive, fluent and intuitive interaction. On the other hand, it is noteworthy, that stemming from the context of a standard, it is not possible to define a conclusive list of requirements. There can be a multitude of unexpected and unintended workflows which may not be thought off. The usage of FHIR resources and especially the FHIR Search Framework is open to arbitrarily different use cases (e.g., Public Health, Medical Intelligence).

As a result, the second research question posed in this thesis was designed to find measures for evaluating non-functional aspects of the FHIR Search Framework and to formulate ideas for supporting stakeholders in general when searching in FHIR data.

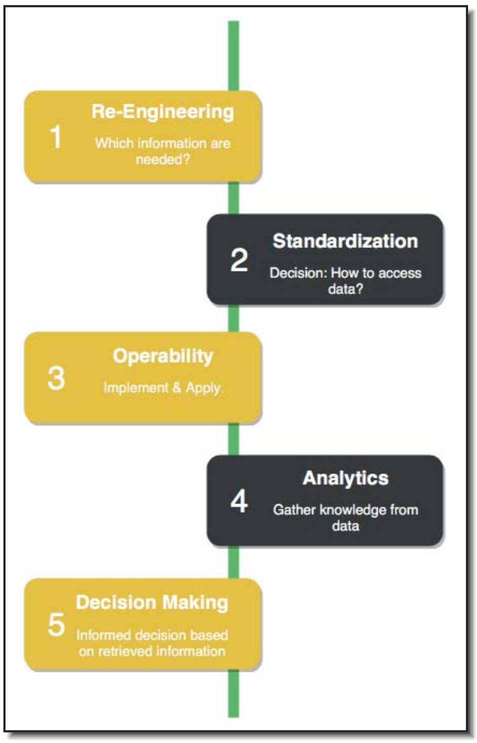

1
Re-Engineering
Which information are needed?
2
Standardization
Decision: How to access data?
3
Operability
Implement & Apply.
4
Analytics
Gather knowledge from data
5
Decision Making
Informed decision based on retrieved information

Figure 23: Workflow: Searching in Health Care Data

When analyzing some of the well-known workflows in which the FHIR Search Framework plays a central role, it is noticeable that the individual steps that are performed in order to get meaningful insights into the health data by searching and filtering are identical to a large degree between stakeholders. The exact questions posed by each individual user may differ, but the overall theme does not vary to a large extend. The expected workflow is shown in Figure 23.

In particular it is noteworthy that stakeholders following these steps are influenced by the non-functional quality aspects gathered in section 4.7. A mapping between the individual tasks and the quality measures can be established:

- "Which information are needed?" is influenced by Appropriateness recognizability
- "How to access data" is influenced by Learnability
- "Implement & Apply" is influenced by Operability

As a consequence, ways to support and foster these quality measures, as well as the reasoning behind the mapping above are discussed in separate sections below.

The steps "Analytics" and "Decision Making" are excluded, because of their subjective nature.

5.1. Appropriateness Recognizability: Documentation

From the perspective of the standards authors, "Appropriateness recognizability" should form the focus when discussing actions for increasing the adoption of the FHIR Search Framework. Without achieving a suitable degree in this category, it becomes difficult to foster interoperability when searching in health care data, due to the fact that appropriateness recognizability is the foundation of all external usability measures. Failing to convince stakeholders, by not conveying which needs are resolvable through the FHIR Search Framework, blocks the way (in worse case scenarios) for innovations through withholding the implementation of interoperable solutions. Therefore, learnability and operability play only a limited role as they do not trigger an investment in the evaluation of achieving interoperability. Challenges with interoperability are likely to occur if data is being exposed through FHIR, without recognizing that advantages may vary depending on the degree to which the search functionalities are implemented. Not all uses cases may be properly supported by simple GET and BATCH operations. Appropriateness Recognizability offers answers to the question whether or not a certain operation is executable (developer perspective) or if certain information can be accessed (user perspective). Thus it is influencing the question of the re-engineering workflow step, due to specifying the general limitations of the use case.

When comparing the costs of implementations and any potential advantages, it is crucial to know which limitations and their impacts are to materialize itself first. To put this goal into practice, guidance should be extended on how the different operations, interactions, search parameters and modifiers interact to form a freely adaptable framework. Due to the structural limitations, such a task is not directly solvable by a standard. However, it can still lay the foundations by depicting the broad range of requests which can be exchanged. It is essential that FHIR highlights the importance the importance of searching, as it can easily be misinterpreted as an "add-on".

Currently, the limitations of the FHIR Search Framework are well acknowledged and discussions are held to which extend it is reasonable to add more complexity, but enable new interactions. The current thesis tries to contribute this discussion by providing an overview of the gains of interoperability and theoretical limits of the API.

Appropriateness recognizability is yet to be improved by presenting the cohesiveness of the FHIR Search Framework, its power, and expressiveness. Individual elements of it, especially advanced aspects, should not be interpreted as "fallback solutions". Implementers should realize that in a broad set of use cases, these operations may simplify the handling of the health care data and should be considered coequally, given their tight integration with the rest of the FHIR API and the data model.

5.2. Learnability: FHIR Search Proficiency

An important factor in conjunction with understandability of the FHIR standard is identifying individual knowledge gaps. This serves two purposes:

- From the perspective of the standard's authors, having an overview of which parts of the specification contain the least understood sections is valuable as it shows hidden potential for improvements.
- Implementers benefit from being able to get an understanding of their proficiency. Based on this information, they can improve their knowledge in certain specialized areas.

To achieve the listed goals, a quiz about the FHIR Search Framework is proposed below. For each feature and section the the specification of the Search API, questions are formulated to test if the respective concept can practically be applied by a user. The questions are designed to be independent. The tests only target a specific area of functionality, but due to the interconnectedness of the search request elements, overlaps can occur.

To foster a learning curve of the participants, the test itself is designed in an adaptive way. For each search concepts a question with a high and low complexity is posed. Hereby, areas of improvements can be determined and improved systematically without getting a feeling of disappointment. The assessment of the complexity of the question is based on the number of individual elements of the FHIR Search Framework, which need to be combined in order to retrieve the correct result set. Reasons for the chosen level are provided for requests classified as "High complexity".

All searches are to be resolved within a single request. For the sake of traceability, potentially correct examples of search requests are provided within this thesis. The exact modality of the quiz, with regard to a correct and suitable evaluation, is still subject to research, but out scope for this thesis. Implementation would need to provide manually curated and immutable test data. Additionally, multi-client functionality would need to be assured.

5.2.1. Quiz: Search Interactions

Question: Search for all resources with the id “example”.
Goal: Searches on all resources
Complexity: Low
Search Request: [base]/?_id=example

Question: Search for all resources which conform to a profile.
Goal: Searches on all resources
Complexity: High (Concept of a profile and a base search need to be understood)
Search Request: [base]/?_profile:missing=false

Question: Search for all resources which have any reference to the patient with the id “example”.
Goal: Searches on a compartment
Complexity: Low
Search Request: [base]/Patient/example/*

Question: Search for all Observations and Encounters of the Patient with the id “example”.
Goal: Searches on a compartment
Complexity: High (_type needs to be combined with a compartment)
Search Request: [base]/Patient/example/*?_type=Observation,Encounter

Question: Search for all ValueSets.
Goal: Searches on a specific resource
Complexity: Low
Search Request: [base]/ValueSet/

Question: Scenario: You - as a General Practitioner - want to retrieve the last MedicationRequest that was entered in your system.
Goal: Searches on a specific resource
Complexity: High (_sort in combination with _count)
Search Request: [base]/MedicationRequest?_sort=-_lastUpdated&_count=1

5.2.2. Quiz: Search Types, Prefixes and Modifiers

Question: Search for all Encounters which took longer than 3 days.
Goal: Searches on Number
Complexity: Low
Search Request: [base]/Encounter?length=gt3

Question: Searches on for a Risk assesment within in the range [0.35 ... 0.955).
Goal: Search on Number
Complexity: High (Concept: Implicit ranges)
Search Request: [base]/RiskAssesment?probability=ge0.35&probability=lt0.955

Question: Searches for all Patients which where born the the year 1984.
Goal: Search on Date
Complexity: Low
Search Request: [base]/Patient?birthdate=1984

Question: Search for all Patient which died on 2000-01-02 (UTC).
Goal: Search on DateTime
Complexity: High (Date Formatting in Search Requests)
Search Request: [base]/Patient?death-date=2000-01-02Z

Question: How can interoperability for searching be fostered when storing a Patient with the name "ÁSTRÍÐUR"?
Goal: Search on String
Complexity: Low
Answer: Enable a search on only the base characters of the name.

Question: Get a list of all Patients with a name starting with A-D, sorted in a lexicographical order.
Goal: Search on String
Complexity: High (Multiple values for a single search parameter)
Search Request: [base]/Patient?name:contains=A,B,C,D&_sort=name

Question: Search for all Conditions that were diagnosed based on a CT scan (Conditions contains evidence of a CT scan).
Goal: Search on Token
Complexity: Low
Search Request: [base]/Condition?evidence:text=CT

Question: Search for all Observations with a code indicating some kind of development disorder.
Goal: Search on Token
Complexity: High (Subsumption Concept)
Search Request: [base]?Observation=code:below=http://snomed.info/sct|5294002

Question: Given an American patient record with a recored height of 6.07 feet, search on it using European measuring units.
Goal: Search on Quantity
Complexity: Low
Search Request: [base]/Observation?value-quantity=gt185||cm

An example for a search on a quantity that would significantly differ from the last example could not be formulated, as there are not a variety of standard search parameters of type Quantity.

5.2.3. Quiz: Advanced Search Requests

Question: Search for all Observations which contain the words "Vital Signs".
Goal: Search with _text
Complexity: Low
Search Request: [base]/Observation?_text=Vital Signs

Question: Search for all male patients which are older than 18 years or all female patients.
Goal: Search with _filter
Complexity: Low
Search Request: [base]/Patient?_filter=(gender eq male and birthdate co 2000) or (gender eq female)

Question: Search for all Observations and Conditions which contain Body weight related SNOMED CT codes.
Goal: Search with _filter
Complexity: High
Search Request: [base]/?_type=Observation,Condition&_filter=code ss 363804004 or code eq 301336003

Question: Search for all intakes of a Medication which were completed. The taken medication should be a medication which was issued via a prescription for the Patient named Jane.
Goal: Chaining
Complexity: Low
Search Request: [base]/MedicationAdministration?prescription.patient.name=Jane &status=completed

Question: Search for all Procedures which were performed in some Encounters due to a Diagnosis asserted by a Practitioner.
Goal: Chaining
Complexity: High (Check if resources in chain exist with _id)
Search Request: [base]/Procedure?context:Encounter.diagnosis:Condition.asserter :Practitioner._id:missing=false

Question: Search for Patients who where involved in some Encounters which were scheduled due to an appointment.
Goal: Chaining
Complexity: Low)
Search Request: [base]/Patient?_has:Appointment:actor:_has:Encounter: appointment:_id:missing=false

Question: Search for Patients who where involved in some Encounters which were scheduled due to an appointment and who at least one DiagnosticReport based on some Observations.
Goal: Chaining
Complexity: High (Reverse Chaining with _filter due to AND combination)
Search Request: [base]/Patient?_filter=(_has:Appointment:actor:_has:Encounter :appointment:_id:missing=false) and (_has:Observation:patient:_has: DiagnosticReport:result:missing=false)

Composite Search Parameters are left out of the quiz as they are just a combination of the search parameter types above.

5.3. Operability: Tool Support

In the current thesis, it was detailedly reported how search parameters form an abstraction level above the resources. From a developers perspective such an approach is necessary to cope with the variety of resource elements, in order to guarantee sufficient performance while retaining flexibility.

Criticism may occur from two perspectives:

- **Performance**
  Even when indexing solely the standard search parameters, performance issues may arise, especially when indexing large datasets. Future evaluations are necessary to determine how to handle these challenges efficiently. However, performance-related discussions are out of the scope of this thesis.
- **Usability**
  It is at least questionable if the chosen form of abstraction can be well understood by users of the FHIR Search Framework. This issue is being accelerated by the fact that there is a discrepancy between the complexity of use cases and the complexity of the search requests. As demonstrated on the basis of Figure 20, even in use cases involving a small number of resource types, it may quickly be necessary to fallback to using advanced search parameters including reverse chaining and _filter to get the wanted information.

The task at hand is now to prepare an environment tailored for specific users which enables to search for relevant information and solve information needs while taking individual capabilities and knowledge levels into account. All users need are to be supported, in combination with exposing search features of the FHIR Search Framework using an appropriate interface. This requirement is valid regardless if a data analysts in a hospital are retrieving information or if Patients want to access their health care conditions in-depth. Unique ways for supporting each stakeholder group should be created to optimally use the foundation offered by FHIR.

It remains an open question how exactly operability can be enhanced as a non-functional quality factor. The current thesis suggest to pursuit an evaluation of using graphs as the foundation for selecting resource elements in conjunction with a visual searching paradigm. Based on the foundational work presented in 3.6, it is argued that such a model may be quite effective in cases where the underlying information model of FHIR is understood by the user (e.g., by developers and data engineers).

Due to the involved complexity of the queries it is furthermore suggested that when implementing the FHIR Search Framework, a test-driven approach should be used. Services are currently missing for "grading" the quality of implementations, based on the correct interpretation of search requests. Only with constant oversight it becomes possible to retain interoperability.

6. Querying Health Care Data

As indicated by the title of the current thesis, it was planned to analyze options for querying health care data in addition to the discussion about searching. Due to the initially unexpected outcome of being able to express a vast variety of search request using the FHIR Search Framework, the priority of finding ways to bridge occurring gaps was reduced. With regard to organizational constraints, it was chosen to only present an excursus into the topic of more sophisticated query languages. Still, due to its overall significance, this subject should be revisited in a dedicated way, as no answers will be given in the current thesis to the following questions:

- At which level of expressiveness do the topics of searching and querying converge?
- How can searching in FHIR and querying health care data complement each other?

This section portraits a prototypical implementation of a Big Data solution for querying health care data. More specifically, it is demonstrated how to perform aggregation queries on top of large data sets of patient data including observations, conditions, and medication information.

For the task at hand, it was decided to build upon the work of the open-source project Bunsen - a library for analyzing FHIR resources in Apache Spark [122]. Apache Spark is a general-purpose computing engine for running data processing jobs in a large scale and distributed environment. An introduction and in-depth description of provided APIs can be found in [123].

From a general point of view, the Spark engine was designed to run computations on data structures in parallel and in-memory. At its core, Spark is responsible for scheduling, distributing and monitoring tasks provided by higher-level APIs. To increase the efficiency for implementing practical use cases, additional libraries can be plugged into Spark. Officially distributed packages provide functionalities around graph processing, machine learning, and data streaming.

Data from various data sources is represented in Spark abstractly by a concept called Resilient Distributed Datasets (RDDs). A RDD is a fault-tolerant and immutable set of data elements which can be accessed in parallel. Moreover, it can be spread out across different Spark nodes. On top of RDDs, it is possible to construct another form of data representation called "Datasets". A Dataset can leverage all features of a RDD while providing type information about itself at compile time to the program executing the analytic queries. This feature is achieved by enabling to impose a data schema. A Dataset is organized in rows and columns, hence it can be accessed in a relational database-like style.

The goal of the aforementioned library Bunsen is to provide RDDs and Datasets of FHIR resources. In combination with offering support for ConceptMaps and advanced ValueSet operations, Bunsen allows importing static FHIR Bundles into Apache Spark. Internally, so-called Encoders are being used to transform the input data into the tabular format of Datasets, which is highly optimized for analytical purposes and stored in a binary format.

Subsequently, SQL queries can be run against a Dataset. To preserve the Datasets for future analysis, they can be stored on disk in an Apache Hive metastore. Apache Spark can read the data natively back into Datasets.

One essential aspect that is often needed in health care, but is currently not being specified by FHIR is to achieve an aggregation view of the health care data. This topic is exemplarily chosen to test the capabilities of Apache Spark and Bunsen. In Listing 1 a query for retrieving the most common observations is presented. In this case, "Observation" is a Hive database populated with synthetically generated health care data. Similar to this approach, arbitrary queries for all kinds of other examples can be formulated using the complete expressive power of SQL.

Listing 1:
Example Apache Spark SQL Query

```
spark.sql("SELECT code.coding.display, count(*) from
    Observation GROUP BY code").orderBy(desc("count(1)")).
    show();

+-------------------+--------+
|            display|count(1)|
+-------------------+--------+
|      [Body Height]|   65773|
|      [Body Weight]|   65773|
|   [Blood Pressure]|   65773|
|  [Body Mass Index]|   58540|
|[Hemoglobin A1c/H...|   49395|
+-------------------+--------+
Only showing top 5 rows
```

In general, Bunsen provides support for reading FHIR Bundles, FHIR resources encoded in JSON or XML, and objects of FHIR resources when using the Java Reference implementation (HAPI FHIR [124]). In cases where a FHIR server does not use the previous mentioned FHIR implementation as its foundation, a gap presents itself. When trying to use Bunsen to achieve a real-time view of the data, each resource would need to be serialized and loaded again from disk into Spark.

As a workaround, until native support for writing to a Hive database can be added to such a server, it is recommended to set up a FHIR Subscription[18] that forwards all newly posted FHIR resources to a middleware service. An example of such a subscription can be found in Appendix G. Afterwards, Bunsen can be used in the middleware service to create Datasets from the received resources. Spark, therefore, essentially contains a mirror of all FHIR resources. Yet, by storing the data in a binary format, no storage-related issues arise. An overview of the used architecture can be found in Figure 24. If the FHIR server executing the subscription is handling the task efficiently enough, a near real-time solution can be achieved.

In the future, a complete and formal evaluation of the performance and limitations of Apache Spark in combination with Bunsen should be performed.

[18] The concept of a FHIR Subscription is explained in [125]

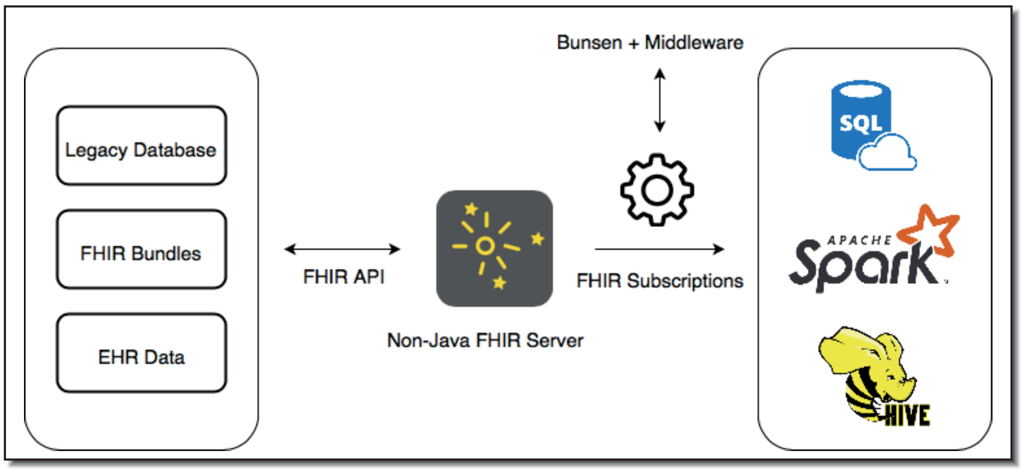


Figure 24: Using Bunsen in combination with FHIR Subscriptions

7. Conclusion

7.1. Final Considerations

Effective searching, querying, filtering, and aggregation - in the context of health care, all of these operations may provide enormous potential, e.g., for medical researchers and users of an EHR. In the end, all involved stakeholder should be enabled to gain insights through harmoniously combining and analyzing data from different systems and institutions. The objectives of innovative projects, products and services in this domain should foster topics around patient empowerment, clinical safety, and improved patient satisfaction.

As a perquisite, interoperability needs to be achieved on a syntactical, semantical and organizational level. The current thesis, focused on outlining how to accomplish this goal by implementing the FHIR Standard. It was discussed how its data model in the form of so-called resources provides clear representations of often needed entities and concepts of health care. Additionally, it was examined how verifiability of the exchanged information becomes feasible through terminology service functionality and the description of organization-specific restrictions in rigid, but machine-readable profiles.

When discussing the above-mentioned topics around data analytics, it becomes noticeable that, e.g., searching and querying seem closely related or even alike, due to the fact that the individual processes behind these terms are intertwined. Moreover, a variety of solutions for implementing them in health care, an environment with steadily growing data requirements, are described in literature. With regard to the main specialization of this thesis, it was shown in-depth how to search and filter using functionality offered by FHIR.

Based on this theoretical foundation, the main goal of the current thesis was set out to answer research questions, which first and foremost were posed to investigate possible open challenges regarding searching in FHIR. This concrete main motive was selected to a) explore potential restrictions of the chosen approach and identify occurring gaps and b) analytically present how searching in FHIR fits together with more advanced concepts like querying and aggregation.

This analysis started out with the initial idea of providing conceptualizations of the main terms, such that semantical differences and a common understanding could be established. The introductory definition of searching incorporated the concept of an information need, based on which stakeholders may mainly act when trying to retrieve information.

Subsequently, a description of the functionality encapsulated within the FHIR Search Framework was provided. The corresponding section aims at being a guide for the specified features in the standard. At first, the complexity was broken down into manageable units by clustering the content of the Search API into interrelated parts. For each of the emerging categories, an in-depth description was provided. Features concerning all search interactions, definitions of search parameters, search parameter types and advanced search concepts were discussed.

Furthermore, a "FHIR Search Components Model" was proposed in order to abstractly describe the underlying information need behind a search request. This model enables to capture how all the examined elements of the FHIR Search Framework can be used in combination. Also, it presents which semantical statement is expressed without relying on a concrete syntax, which may still change in a future revision of the FHIR standard.

Based on the discussed research, one main paradigm regarding searching was identified in FHIR: "Finding resources in a haystack of information". Resources which shall be included by an executing system as a result can be described through search parameters, as well as, values and restrictions for each of these parameters. Moreover, structural information, like the interconnectedness of resources, can be inquired likewise. Via a RESTful API, all of these aspects are passed to a FHIR-enabled Server, which determines appropriate matches and returns these in a FHIR bundle. Resultantly, the user is responsible for selecting properties of FHIR resources, which describe at best the information that is needed, i.e., the user's information need.

A conclusive answer to the first research question posed in this thesis regarding the maturity of the API was given in chapter 3 and 4. The following main contributions were discussed:

- **Quality Measures for the FHIR Search Framework**
  A set of measures for the effectiveness of the FHIR Search Framework was derived in a systematic way based on the notion of quality. Inspired by ISO 25010, concrete metrics were adapted for the evaluation of search requests.

- **Current Limitations / Indications for future research**
  For each category of functionality offered by the standard, an assessment of completeness with regard to the previously established quality measures was performed. As a consequence of this analysis, a total of eleven change request for the FHIR standard itself have been submitted. A list of these changes can be found in Appendix F. The changes itself can be classified as follows: three instances of underspecifications were improved, four minor corrections of examples were submitted, and three improvements concerning the handling of advanced features like _filter were proposed.

  Yet, the maturity of the FHIR Search Framework was proven, as mostly non-substantial issues where discovered. Only a single change request was marked as substantial. In most cases, completeness within the self-given boundaries could be shown. Having said this, a few optimization potentials were discovered, all of which are solely concerning more complex use cases, thus underlining the quality of the FHIR Search Framework.

  It is highlighted that besides using the quality measures to asses the completeness of each functional category, a graph model describing the interconnectedness of FHIR resources was introduced. Using graph expressiveness models, the functionality regarding chained search requests was studied. From a theoretical perspective, _filter in combination with reverse chaining and chaining is considered to provide broad coverage of possible operations in a graph search for determining reachability of two nodes (mirroring the functionality of chained searches in FHIR). The exact breakdown of the search requests, which are formally shown to be inexpressible, can be found in Table 5 and 6.

All in all, from a functional perspective the FHIR Search Framework can be considered well-suited. Practical limitations should be minimal, grounded on the fact that extensive coverage of the lowest expressiveness classes, RPQs and 2RPQs, can be achieved. In limited cases, search requests are not describable by the regular FHIR Search Framework. However, _filter can be used supplementarily. Recursive RPQs are the only exception in this regard. Severe gaps where identified in the support of C(2)RPQs. It is to be evaluated in the future if the REST API is to be extended to encourage these more complex requests. Such a decision is to be based on the fundamental design principles of FHIR.

As presented in a transparent manner in section 1.2, the first research question was concerned with the quality of the FHIR Search Framework. In addition to the functional aspects, quality measures for usability were identifies as well. As depicted through their definitions, relevancy is given, especially when considering developers, which are the primary audience of the FHIR Standard.

Based on Appropriateness recognizability, Learnability, Operability, ways of empowering users and developers and to better support the transformation of an information need into a search request were designed. As the main contribution, a quiz helping to learn how to search in FHIR is highlighted. Finally, an excursus into use cases requiring more sophisticated capabilities was provided based on the FHIR-nativ querying library Bunsen.

As an overall result, supporting arguments for the main theme of this thesis were practically validated: it is insufficient to label FHIR only as an underlying technology layer. Its potential strength, does not come from its syntax or any other defined capabilities, but from the general possibilities for the interactions and integration it enables. Through meaningful information exchanges, it builds bridges between systems and organisation.

7.2. Future Work

The evaluation of the FHIR Search Framework showed promising results in terms of functional completeness. Yet, the standard is still evolving and certain parts of the Search API are neither well-known nor implemented widely. In the future, it becomes necessary to validate the theoretical results of this thesis using real-world use cases and working implementations. In addition, another open question arises. Based on the gathered list of syntactical structures which currently cannot be expressed in a graph search, it should be discussed if support for these features is to be added in the FHIR standard, respectively, how to achieve them otherwise. The trade-offs (complexity vs. expressiveness) of these potential additions are to be reviewed by the FHIR community. If a positive answer is given, it is to be answered how they could be "optimally" integrated. For example, it is to be considered if _filter should be pushed more as an alternative and be incorporated as an approach offering more flexibility.

A complete topic which has not been considered in literature, to the best of the current author's knowledge, is how to achieve interoperability when searching across data-models of different standards. Lastly, it is empathized that interoperability is only given to a suitable degree if specifications are implemented correctly. It is to be investigated if a coherent testing framework for searching in FHIR resources can alleviate pain points around this issue.

A. Appendix: Examples RESTful FHIR API Interactions

Please note that for all of the following examples any FHIR-enabled server can be used, provided that the server generally supports the described RESTful interactions.

To create a valid FHIR API request, [base] should be substituted with the correct Service Base URL (e.g., https://test.fhir.org/r3).

Responses from the FHIR server are not documented as they are subject to individual server behaviour and the stored resources at the respective time.

A.1. Capabilities Interaction

Listing 2:
Capabilities Interaction

```
1. GET [base]/metadata
```

1. Get the server's Capability Statement, which describes all of the supported API interactions and FHIR resources.

A.2. Read Interaction

Listing 3:
Read Interaction

```
1. GET [base]/Patient/example

2. GET [base]/Observation/example?_format=json

3. GET [base]/DiagnosticReport/example?_summary=text
```

1. Get the FHIR resource of type "Patient" containing the logical id "example".
2. Get the FHIR resource of type "Observation" containing the logical id "example". Explicitly indicate that a response encoded in JSON is expected.
3. Get the FHIR resource of type "DiagnosticReport" containing the logical id "example". Return only the "text" element of the selected resource.

A.3. Delete Interaction

Listing 4:
Delete Interaction

```
1. DELETE [base]/Patient/example
```

1. Delete the FHIR resource of type “Patient” containing the logical id “example”.

A.4. Create Interaction

Listing 5:
Create Interaction

```
1.  POST [base]/Patient

{
  "resourceType": "Patient",
  "text": {
    "status": "generated",
    "div": "<div>Generated Narrative...</div>"
  },
  "active": true,
  "name": [
    {
      "use": "official",
      "family": "Chalmers",
      "given": [
        "Peter",
        "James"
      ]
    }
  ],
  "managingOrganization": {
    "reference": "Organization/1"
  }
}
```

1. Create a new Patient FHIR resource for the fictional patient Peter James Chalmers. This patient record is actively being managed by the organization containing the logical id 1 within the scope of the current FHIR server.

A.5. Update Interaction

Listing 6:
Update Interaction

```
1.   PUT [base]/Patient/example

{
  "resourceType": "Patient",
  "id":"example",
  "text": {
    "status": "empty",
    "div": ""
  },
  "active": true,
  "name": [{
      "use": "official",
      "family": "Chalmers",
      "given": [
        "Peter"
     ]}
  ],
  "birthDate": "1974-12-25",
  "_birthDate": {
    "extension": [{
        "url": "http://hl7.org/fhir/StructureDefinition/
           patient-birthTime",
        "valueDateTime": "1974-12-25T14:35:45-05:00"
     }]
  },
  "managingOrganization": {
    "reference": "Organization/1"
  }
}
```

1. Update the Patient resource created in Listing 5. Add a birthDate field, as well as an extension containing the exact DateTime of the birth.

A.6. History Interaction

Listing 7:
History Interaction

```
GET [base]/Patient/example/_history
```

1. Retrieve a bundle containing all versions of the FHIR Patient resource with the logical id “example”.

B. Appendix: EHR Statistics University Clinics Netherlands - Volume

B.1. Statistics VU University Medical Center Amsterdam

Name	Row Count	Data Space Used (KB)	Data Space used (KB) per Entry
DCM_Verrichting	78459012	50378464	0,642
DCM_Lab	64007779	19110272	0,299
DCM_Afspraak	7820949	3111568	0,398
DCM_MedischeDiagnose	6942551	2719656	0,392
DCM_Opname_Opnameperiode_PerBed_PerDag	6599646	6125656	0,928
DCM_Medicatie_Voorschrift	6444260	3849080	0,597
DCM_Medicatie_Toediening	4063104	909624	0,224
DCM_Patient_koppel	2196963	123800	0,056
DCM_ParamedischeZorg	2173883	441672	0,203
DCM_Patient	2073994	593264	0,286
DCM_Opname_Opnameperiode_PerBed	1768116	1933928	1,094
DCM_Opname_Opnameperiode	1702464	1718832	1,01
DCM_DBC_eigenaar	1434846	240080	0,167
DCM_DBC	1434846	1455128	1,014
DCM_AfspraakOrder	1407696	484848	0,344
DCM_Opname_Opname	1354911	837992	0,618
DCM_Lab_MMI_BepalingenTekst	1245140	384144	0,309
DCM_Lab_MMI_Resistentie	1202274	156552	0,13
DCM_Meting_Circulatie_Hemodynamiek	916751	389088	0,424
DCM_Meting_BMI	867224	202992	0,234
DCM_Meting_Respiratie_Algemeen	684296	170256	0,249
DCM_Biomateriaal_Sample	577814	97624	0,169
DCM_Meting_Gewicht	531323	90096	0,17
DCM_LDA	509956	258472	0,507
DCM_Meting_Pijnbeoordeling	480311	41872	0,087
DCM_Meting_Lengte	459631	81768	0,178
DCM_Voorgeschiedenis_Medisch	345398	46152	0,134
DCM_OK_Verrichting	344612	279048	0,81
DCM_Problemlist	324613	51552	0,159
DCM_OK_Zitting	298030	223368	0,749
DCM_Meting_Rookgedrag	268356	33136	0,123
DCM_Lab_MMI_BepalingenGetal	227930	68376	0,3
DCM_Meting_Respiratie_Beademingsmeetwaarden	221929	145168	0,654
DCM_Meting_SNAQ_Score	202857	26400	0,13
DCM_Meting_Circulatie_Temperatuur	200892	46040	0,229
DCM_Voorgeschiedenis_Chirurgisch	194431	21144	0,109
DCM_Meting_Respiratie_Beademingsinstellingen	193012	98904	0,512
DCM_Flowsheet_antwoorden	180427	81096	0,449
DCM_Lab_MMI_Isolaten	161873	47376	0,293
DCM_Meting_Neurologie_Convulsie	144764	31120	0,215
DCM_Flowsheet_variabelen	143309	82760	0,577
DCM_Meting_Neurologie_Uitval	122692	53928	0,44
DCM_Meting_Neurologie_GlasgowComaScaleScore	117850	30768	0,261
DCM_VochtOutput	116760	13592	0,116
DCM_Hoofdbehandelaar	79777	12976	0,163
DCM_Meting_DelierScreening_score	62045	8248	0,133
DCM_Voorgeschiedenis_Familie	54176	4288	0,079
DCM_Meting_Respiratie_BronchiaalToilet	45712	18552	0,406
DCM_Meting_Bradenschaal_score	45326	11168	0,246
DCM_ResearchStudy_Deelname	44356	10728	0,242

DCM_Meting_DOSS_score	31442	15760	0,501
DCM_Meting_JohnsHopkinsValrisico_score	29978	8040	0,268
DCM_Meting_KatzADL6_score	29393	7368	0,251
DCM_Meting_Decubitus_score	28273	4320	0,153
DCM_Meting_Nefrologie_DialyseInstellingen	24655	14424	0,585
DCM_Meting_Nefrologie_DialyseMeetwaarden	24612	10984	0,446
DCM_Meting_Neurologie_Intracraniele_Druk	11155	1816	0,163
DCM_Meting_Neurologie_Cerebrale_Perfusieve_Druk	10090	1728	0,171
DCM_VochtInput	6309	800	0,127
DCM_Infectie	5789	1576	0,272
DCM_Meting_RASS_Score	5615	504	0,09
DCM_Meting_Circulatie_Thermoregulatie	5445	1344	0,247
DCM_Isolatie	5020	680	0,135
DCM_Meting_BPS_score	4370	1120	0,256
DCM_Meting_Circulatie_AanvullendeMonitoring	4220	1744	0,413
DCM_OK_Zitting_Afgezegd	3962	2320	0,586
DCM_Meting_Circulatie_Algemeen	3955	864	0,218
DCM_Lab_MMI_BepalingenTiter	1568	488	0,311
DCM_Meting_Circulatie_ExternePacemaker	1270	648	0,51
DCM_ResearchStudy	1016	288	0,283
DCM_Meting_Circulatie_ECMO	976	704	0,721
DCM_Meting_CAMICU_score	954	240	0,252
DCM_Meting_Circulatie_PICCO	764	352	0,461
DCM_Meting_Circulatie_IABP	522	208	0,398
DCM_Meting_Neurologie_Lumbaaldruk	277	48	0,173
DCM_Meting_Respiratie_Weanen	228	88	0,386

B.2. Statistics Leiden University Medical Center

Name	Row Count	Data Space Used (KB)	Data Space Used (KB) per
DCM_Meting_Metavision	3436822051	484450552	0,141
DCM_Meting_Generiek	686939802	147546152	0,215
DCM_Order_Statusmutaties	306009378	38151856	0,125
DCM_LAB_Bepaling	90623825	36454592	0,402
DCM_Meting_Bloeddruk	154886286	31346672	0,202
DCM_Rontgen_Verslag	5309062	7394312	1,393
DCM_Medicatie_Toediening	18659038	5931552	0,318
DCM_Contact_AgendaAfspraak	14574806	5247528	0,36
DCM_Order_Order	16922870	4548784	0,269
DCM_Medicatie_Voorschrift	9169686	4153576	0,453
DCM_Order_Orderinhoud	23140652	3803984	0,164
DCM_DCR_Generiek	4601289	2206360	0,48
DCM_DCR_Diagnose	4197723	2010336	0,479
DCM_LAB_Aanvraag	10219378	1412672	0,138
DCM_DBC	4658171	1167456	0,251
DCM_Rontgen_Verrichting	6658352	852248	0,128
DCM_Rontgen_onderzoek	5273274	774696	0,147
DCM_Contact_Opname_Periode	2542457	585872	0,23
DCM_VochtBalans_Totaal	4271555	422968	0,099
DCM_Contact_Opname_Volledig	1395772	384080	0,275
DCM_Contact_OK	514363	284176	0,552
DCM_Contact_OK_Personeel	2081031	244536	0,118
DCM_Contact_OK_Verrichtingen	769692	178808	0,232
DCM_EXT_NEO_Overdracht_Tekst_Uitslag	1279023	158872	0,124
DCM_Contact_OK_Cluster	1094348	156832	0,143
DCM_Meting_Pijn	1332090	156120	0,117
DCM_PatientDossier	1127509	136400	0,121
DCM_EXT_NEO_Medicatie_voorschrift	422093	116592	0,276
DCM_Meting_Gewicht	1273055	70168	0,055
DCM_Meting_BSA	1110620	68392	0,062
DCM_Contact_SEH_Bezoek	345333	66032	0,191
DCM_Pathologie_Test	359319	55576	0,155
DCM_Pathologie_Monster	359319	53696	0,149
DCM_Meting_BMI	1110620	51688	0,047
DCM_DCR_Voorgeschiedenis	123829	47552	0,384
DCM_DCR_Complicatie	101568	31832	0,313
DCM_Meting_Lengte	587582	26944	0,046
DCM_EXT_NEO_Contact_AgendaAfspraak	368841	25792	0,07
DCM_IngebrachteMaterialen	175596	19752	0,112
DCM_VochtBalans_Patient_Per_dag	383984	19480	0,051
DCM_Contact_SEH_Behandelaars	243959	17480	0,072
DCM_Zorgverlener	112566	13544	0,12
DCM_EXT_NEO_Overdracht_Concern	57682	11968	0,207
DCM_Order_Suborderdefinitie	70224	6736	0,096
DCM_Allergie	45239	5976	0,132
DCM_Contact_Opname_PDMS	145142	5720	0,039
DCM_Opname_Mediscore	31681	5072	0,16
DCM_Meting_Delier_DOS	79261	3896	0,049
DCM_Meting_Ondervoeding	54715	1680	0,031
DCM_Contact_OK_GebruikteArtikelen	8111	1544	0,19
DCM_Order_Orderdefinitie	3855	552	0,143
DCM_EXT_NEO_PATIENT	6673	456	0,068
DCM_EXT_NEO_Meting_BMI	6673	312	0,047
DCM_Specialisme	161	16	0,099

Technology
Arts Sciences
TH Köln

C. Appendix: EHR Statistics University Clinics Netherlands - Variety

ZIB	UMCU DCM	Vumc DCM	LUMC	FHIR
Ademhaling				
AlcoholGebruik		Alcoholgebruik		Observation
Alert				Flag
AlgemeneMentaleFuncties				
AlgemeneMeting	Meting		Meting_Generiek	
AllergieIntolerantie			Allergie	AllergyIntolerance
BarthelIndex				
BehandelAanwijzing				Consent
Behandeldoel				
Betaler				Coverage, Organization, Patient
Blaasfunctie				
Bloeddruk	Bloeddruk		Meting_Bloeddruk	Observation
Brandwond				
BurgerlijkeStaat				
Communicatievaardigheden				
Contact	Agenda	Afspraak	Contact_AgendaAfspraak	Encounter
Contactpersoon				RelatedPerson
Darmfunctie				
DecubitusWond				
DrugsGebruik		Drugsgebruik		Observation
Familieanamnese				
FunctieHoren				
FunctieZien				
FunctieZintuiglijkeWaarneming				
FunctioneleOfMentaleStatus				Observation
Gezinssituatie				
GlasgowComaScale				
Hartfrequentie				
Huidaandoening				
HulpBijMedicatie				
HulpVanAnderen				
Infuus				
Levensovertuiging				
Lichaamsgewicht	Lichaamsgewicht	Meting_Gewicht	Meting_Gewicht	Observation
Lichaamslengte	Lichaamslengte	Meting_Lengte	Meting_Lengte	Observation
Lichaamstemperatuur				
MedicatieGebruik				MedicationStatement, MedicationRequest, MedicationDispense
MedicatieToediening		MedicatieToediening	Medicatie_Toediening	
MedicatieVerstrekking				
Medicatievoorschrift	MedicatieVoorschrift	MedicatieVoorschrift	Medicatie_Voorschrift	
MedischHulpmiddel				Device, DeviceUseStatement
Menstruatiecyclus				
Mobiliteit				
MUSTScore				
Nationaliteit				
O2Saturatie				
Ondervoeding			Meting_Ondervoeding	
Opleiding				
OverdrachtConcern	MedischeDiagnose	MedischeDiagnose	DCR_Diagnose	
OverdrachtGeplandeZorgActiviteit				ProcedureRequest, MedicationRequest, ImmunizationRecommendation, DeviceRequest, Appointment
OverdrachtLaboratoriumUitslag	Lab	Lab	LAB_Bepaling / LAB_Aanvraag	Observation, Specimen
OverdrachtTekstUitslag				
OverdrachtValrisico				
OverdrachtVerrichting	Verrichting	Verrichting		
ParticipatieInMaatschappij				
Patient	Patient	Patient		Patient
PijnBeleving				
Pijnscore		Pijnscore	Meting_Pijn	
Polsfrequentie				
Slaapfunctie				
SNAQScore		Meting_SNAQ_Score		
SondeSysteem				
SpecifiekeMentaleFuncties				
Stoma				
Taalvaardigheid				
TabakGebruik		Meting_Rookgedrag		Observation
UitkomstVanZorg				
Vaccinatie				Immunization, ImmunizationRecommendation
VermogenTotEten				
VermogenTotDrinken				
VermogenTotHaarverzorging				
VermogenTotMondverzorging				
VermogenTotToiletgang				
VermogenTotZichKleden				
VermogenTotZichWassen				
VerpleegkundigeInterventie				
Voedingsadvies				NutritionOrder
VrijheidsbeperkendeMaatregelen				
VrijheidsbeperkendeMaatregelenGGZ				
Wilsverklaring				Consent
Wond				
Woonsituatie				Observation
Ziektebeleving				
Zorgaanbieder	Zorgaanbieder			Organization
Zorgverlener	Zorgverlener	Zorgverlener	Zorgverlener	Practitioner, PractitionerRole
Zwangerschap				
	Biomateriaal	Biomateriaal_Sample		
	ECG			
	Echo			
	Opname	Opname_Opname	Contact_Opname_Volledig	
		Opname_Opnameperiode	Contact_Opname_Periode	

	DBC	DBC	DBC
		LDA (lines, drains and airways)	
	BMI	Meting_BMI	Meting_BMI
		Meting_BPS_score	
		Meting_CAMICU_score	
		Meting_Circulatie_AanvullendeMonitoring	
		Meting_Circulatie_Hemodynamiek	
		Meting_Circulatie_Temperatuur	
		Meting_DOSS_score	Meting_Delier_DOS
		Meting_Neurologie_Cerebrale_Perfusieve_Druk	
		Meting_Neurologie_Convulsie	
		Meting_Neurologie_ElectroEncephaloGram	
		Meting_Neurologie_Intracraniele_Druk	
		Meting_Neurologie_Lumbaaldruk	
		Meting_Neurologie_Uitval	
		Meting_Respiratie_Algemeen	
		Meting_Respiratie_Beademingsinstellingen	
		Meting_Respiratie_Beademingsmeetwaarden	
		Meting_Respiratie_BronciaalToilet	
		Meting_Respiratie_Weanen	
	OK _Zitting	OK _Zitting	Contact_OK
	OK_Verrichting	OK_Verrichting	Contact_OK_Verrichtingen
		ParamedischeZorg	
		VochtOutput	
	Bloedtransfusie	Bloedtransfusie	
	Radiologie		Rontgen
	Order		
	SEH		
	Studie/Studiedeelnemer		
	MetingAudiologie		
	RadiotherapieDossier		
	IC		
	Dossier		
	Vragenlijsten	Vragenlijsten	
			DCR_Complicatie
			DCR_Voorgeschiedenis
			Meting_BSA
			Meting_Metavision
			IngebrachteMaterialen
			Opname_Mediscore
			Contact_OK_Cluster
			Contact_OK_Personeel
			Contact_OK_GebruikteArtikelen
			Pathologie_Monster
			Pathologie_Test
			Pathologie_Uitslag
			Order_Order
			Order_Orderdefinitie
			Order_Statusmutaties
			Order_Orderinhoud
			Order_Suborderdefinitie
			Contact_Opname_PDMS
			Contact_SEH_Bezoek
			Contact_SEH_Behandelaars
			PatientDossier

D. Appendix: Examples FHIR Search Requests

Please note that for all of the following examples any FHIR-enabled server can be used, provided that the server generally supports the described Search interaction and the individual search parameters.

To create a valid search request, [base] should be substituted with the correct Service Base URL (e.g., https://test.fhir.org/r3).

Responses from the FHIR server are not documented as they are subject to individual server behaviour and the stored resources at the respective time.

D.1. Search Parameters & Search Context

Listing 8:
Example Search: Common Parameters

```
1. GET [base]/?_id=example

2. GET [base]/?_profile:below=http://hl7.org/fhir/
   StructureDefinition/

3. GET [base]/?_lastUpdated=gt<date>

4. GET [base]/?_content=Text included in the resource
   narrative
```

1. Get all resources that are being managed by the FHIR Server and contain the logical id “example”.
2. Get all resources that claim conformity to a profile that is being officially managed by HL7.
3. Get all resources that were last updated later than <date>.
4. Get all resources that contain the keywords provided by _content in their narrative text.

Listing 9:
Example Search: _type

```
1. GET [base]/?_type=StructureDefinition ,ValueSet&url:
   below=<PartialCanonicalURL>

For example:
GET [base]/?_type=StructureDefinition ,ValueSet&url:below=
   http://hl7.org/fhir/StructureDefinition/
```

1. Get all StructureDefinitions and ValueSets that are currently known by the server and that contain <PartialCanonicalURL> as a prefix of their unique logical URI (Canonical URL).

Listing 10:
Example Search: Compartements

```
1. GET [base]/[compartement]/[id]/[ResourceWith
   OutgoingReference]?[SearchParam]=<value>

2. GET [base]/[compartement]/[id]/*

For example:
GET [base]/Patient/example/Observation?status=final
```

1. Get all Observation resources that contain a reference to the Patient with the logical id “example” and which have the observation status set to “final”.
2. Get all resources that are within the specified Patient compartment.

Listing 11:
Example Search: Resource-specific Parameters incl. OR-Searches

```
1. GET [base]/[resource]?[SearchParam]=<value>&[Search
   Param2]=<value>

2. GET [base]/[resource]?[SearchParam]=<value>,<value2>

For example:
GET [base]/Patient?name=Peter,Vera
GET [base]/Patient?name=Vera&gender=female
```

1. Get all Patients that contain the name Peter or Vera.
2. Get all female Patients that contain the name Vera.

D.2. Search Parameter Types

Listing 12:
Example Search: Token Search Parameters

```
1. GET [base]/Observation?code=29463-7

2. GET [base]/Observation?code=http://loinc.org|29463-7

3. GET [base]/Observation?code=http://snomed.info/sct|
```

1. Get all Observations which are a measurement for a Body weight (LOINC: 29463-7)
2. This search request has the same semantics as above. It additionally specifies the Code System from which the code is drawn.
3. Get all Observations that contain any SNOMED CT codes.

Listing 13:
Example Search: Reference Search Parameters

```
1. GET [base]/Patient?organization=example

2. GET [base]/Observation?subject=Patient/example

3. GET [base]/Observation?subject=[Base]/Patient/example
```

1. Get all Patients which are being managed by the organization with the logical id “example”.
2. Get all Observations which have a reference to a Patient with the logical id “example” in their subject resource element.
3. This search example has the same semantics as above. It differs only by the fact that the reference is explicitly an absolute URL instead of a local relative reference.

Listing 14:
Example Search: Quantity Search Parameters

```
1. GET [base]/Observation?value=le90|http://unitsofmeasure
   .org|kg
```

1. Get all Observations which can be measured in kg and were the value is below 90kg. Please note, as the search parameter value indicates that a UCUM unit it to be used, a search on a canonical value may be executed. This means that, e.g., also values measured in lbs (all Observations below ≈ 198lbs) may be matched.

D.3. Modifiers

Listing 15:
Example Search: Text Modifier

```
1. GET [base]/Organization?name:exact=Firely

2. GET [base]/Organization?name:contains=fhir

3. GET [base]/Observation?code:text=Body temperature
```

1. Get all information relating to an organization with the exact Name “Firely”. Please note that “firely” or any other combination would not be matched.
2. Get all Organizations which names match any combination of the word fhir, e.g., FHIR, fhir, Fhir, FHIR-Test, Test-FHIR.
3. Get all Observations that contain a code which represents the measurement of the Body temperature.

Listing 16:
Example Search: :not Modifier

```
1. GET [base]/Observation?code:not=http://loinc.org|3141-9
```

1. Get all Observations that contain a code which is not a not measurement for “Body weight Measured” (LOINC Code: 3141-9).

Listing 17:
Example Search: :missing Modifier

```
1. GET [base]/MedicationStatement?taken:missing=false

2. GET [base]/Patient?organization.name:missing=true

3. GET [base]/Encounter?_has:Observation:encounter:code:
   missing=true
```

1. Get all MedicationStatements that contain information about the intake of the medication.
2. Get all Organizations that do not contain a name.
3. Get all Encounters that are referenced by Observations that do not include a code for the recorded measurement.

Listing 18:
Example Search: :above,:below Modifier

```
1. [base]/StructureDefinition?url:below=http://hl7.org/
   fhir/
2. [base]/Condition?code:below=http://snomed.info/sct
   |73211009
3. [base]/Condition?code:above=http://snomed.info/sct
   |73211009
```

1. Get all StructureDefinitions which contain a canonical URL which is within the realm of HL7. This search request depicts a :below search on a URL.
2. Get all Conditions that describe children elements of the SNOMED CT Code 73211009 - Diabetes mellitus (disorder). For example, it would match 530558861000132104 - Atypical diabetes mellitus (disorder).
3. Get all Conditions that describe any parent element of the SNOMED CT Code 73211009 - Diabetes mellitus (disorder). It matches parent elements at any level of the hierarchy. For example, it would match a Condition with 20957000 - Disorder of carbohydrate metabolism (disorder) and 75934005 - Metabolic disease (disorder). Please note that Metabolic disease (disorder) is the parent element of Disorder of carbohydrate metabolism (disorder).

Listing 19:
Example Search: :in,:not-in Modifier

```
1. /Observation?category:in=http://hl7.org/fhir/ValueSet/
   observation-category

2. [base]/Observation?category:not-in=http://hl7.org/fhir/
   ValueSet/observation-category
```

1. Get all Observation that contain a code which is included in the recommended ValueSet that is bound to the code element in the FHIR specification.
2. Get the exact complementary set of resource from the example above.

D.4. Advanced Search Concepts

Listing 20:
Example Search: Composite Search Parameters

```
1. [base]/Observation?component-code-value-quantity=http
   ://loinc.org|29463-7$gt80|kg
2. [base]/Group?characteristic-value=gender$mixed
3. [base]/Group?characteristic-value=gender$mixed,
   owner$peter
```

1. Get all Observations that represent a measurement for the given LOINC code with a value greater than 80kg. Please note that this search is equivalent to a search which would result in the intersection of these two criteria, e.g., [base]/Observation?code=29463-7&value-quantity=gt60||kg.
2. Get all Groups that have a characteristic "gender" with a text value of "mixed". Please note that this is not the same as a search which would result in the intersection of gender and its value. In such a case, more results could be matched, e.g., Groups that have the characteristic "gender" and the value "female", but additionally have a value "mixed" for another key.
3. Get all Groups that conform to a set of characteristic: match the gender and the owner characteristic with the given values.

Listing 21:
Example Search: Chaining

```
1. GET [base]/DiagnosticReport?subject:Patient.name=peter
2. GET [base]/DiagnosticReport?result:Observation.based-on
   :CarePlan.identifier=12345
```

1. Get all DiagnosticReports that are referencing a Patient with the name "peter".
2. Get all DiagnosticReports that contain an Observation which in turn contains a reference (basedOn) to a CarePlan with the identifier "12345".

Listing 22:
Example Search: Reverse Chaining

```
1. [base]/Patient?_has:Observation:patient:code=29463-7
2. [base]/Patient?_has:Observation:patient:performer:
   Organization.identifier=1234
3. [base]/Patient_has:Observation:patient:performer:
   Organization.identifier=1234&_has:DiagnosticReport:
   subject:code=1234
```

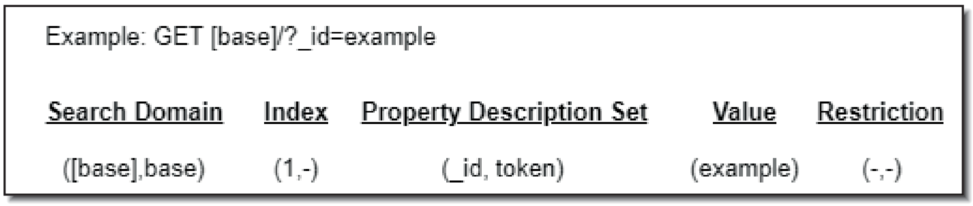
1. Get all Patient resources which are referenced by any Observation through a reference in the "patient" element of the Observation and where the Observation is a measurement of the given code.
2. Get all Patient resources which is referenced by any Observation through a reference in the "patient" element of the Observation and where the Observation additionally contains a reference to an Organization (as a performer). This organization has the given identifier.
3. This result set is a joined set of two independently executed _has queries:
   a) Get all Patient resources which is referenced by any Observation through a reference in the "patient" element of the Observation and where the Observation additionally contains a reference to an Organization (as a performer). This organization has the given identifier.
   b) Get all Patients which is referenced by any DiagnosticReport through a reference in the subject element of the DiagnosticReport and where the DiagnosticReport additionally has the given code as a description.

Listing 23:
Example Search: _filter

```
1. [base]/Patient?_filter=active eq false or ( name eq Eve
    or name eq Peter )
2. [base]/Observation?_filter=patient re Patient/example
   and performer.name ne Todd
```

1. Get all Patients which records are not actively being used and contain a name that is either Eve or Peter.
2. Get all Observations which are referencing a Patient with the logical id "example" and where the name of the Practitioner is not Todd

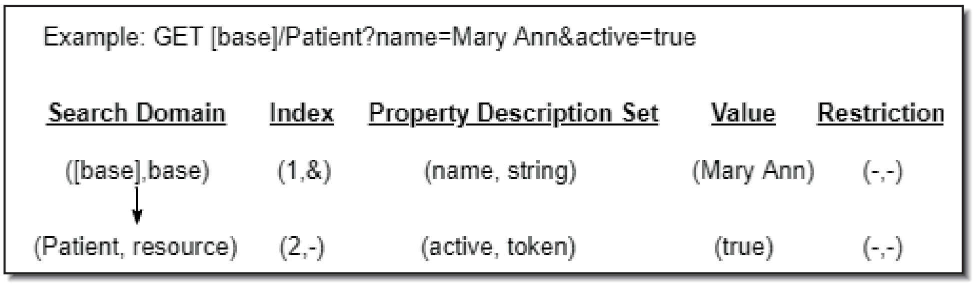
E. Appendix: Examples Search Components Model



Example: GET [base]/?_id=example

Search Domain	Index	Property Description Set	Value	Restriction
([base],base)	(1,-)	(_id, token)	(example)	(-,-)

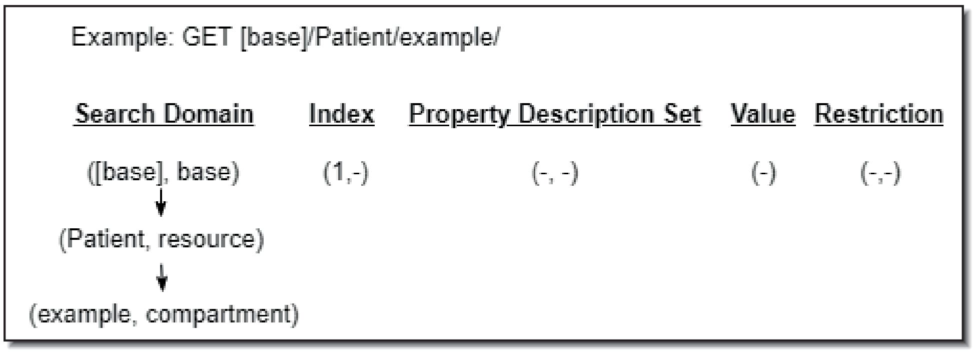
Figure 25: FHIR Components Model: Base Search



Example: GET [base]/Patient?name=Mary Ann&active=true

Search Domain	Index	Property Description Set	Value	Restriction
([base],base)	(1,&)	(name, string)	(Mary Ann)	(-,-)
(Patient, resource)	(2,-)	(active, token)	(true)	(-,-)

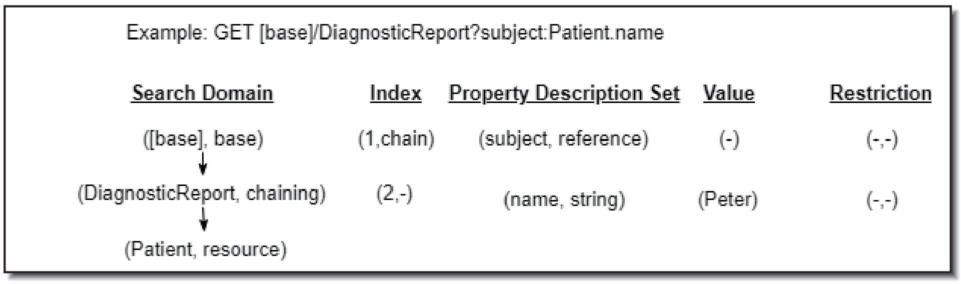
Figure 26: FHIR Components Model: Resource-specific search



Example: GET [base]/Patient/example/

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,-)	(-, -)	(-)	(-,-)
(Patient, resource)				
(example, compartment)				

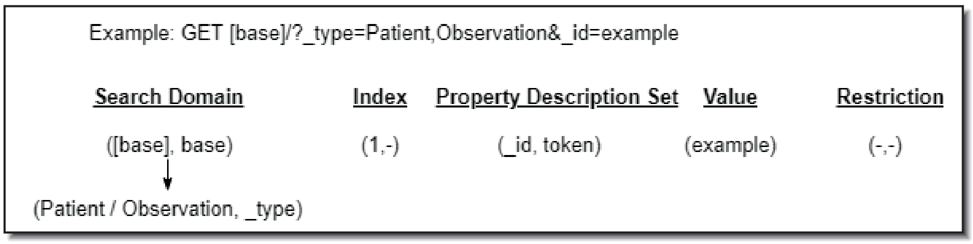
Figure 27: FHIR Components Model: Compartment search



Example: GET [base]/DiagnosticReport?subject:Patient.name

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,chain)	(subject, reference)	(-)	(-,-)
(DiagnosticReport, chaining)	(2,-)	(name, string)	(Peter)	(-,-)
(Patient, resource)				

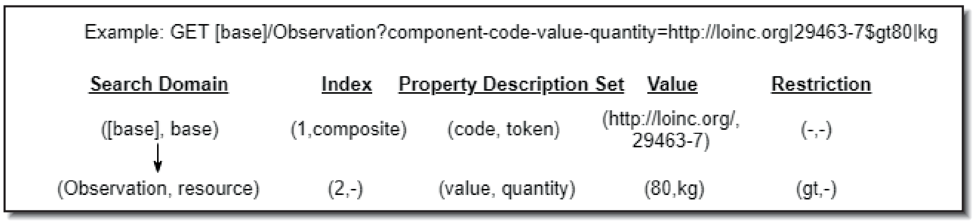
Figure 28: FHIR Components Model: Chained Search



Example: GET [base]/?_type=Patient,Observation&_id=example

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,-)	(_id, token)	(example)	(-,-)
↓				
(Patient / Observation, _type)				

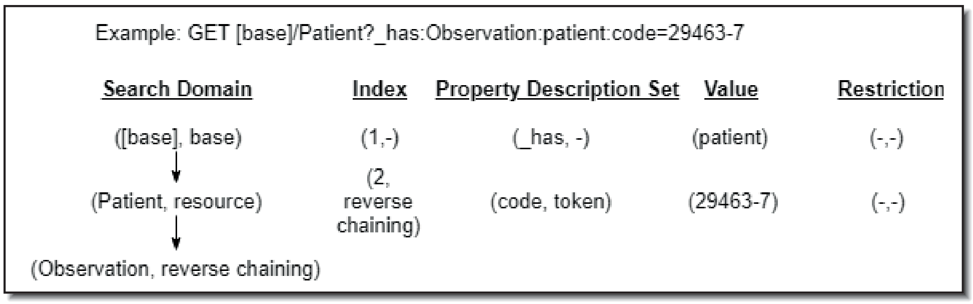
Figure 29: FHIR Components Model: _type



Example: GET [base]/Observation?component-code-value-quantity=http://loinc.org|29463-7$gt80|kg

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,composite)	(code, token)	(http://loinc.org/, 29463-7)	(-,-)
↓				
(Observation, resource)	(2,-)	(value, quantity)	(80,kg)	(gt,-)

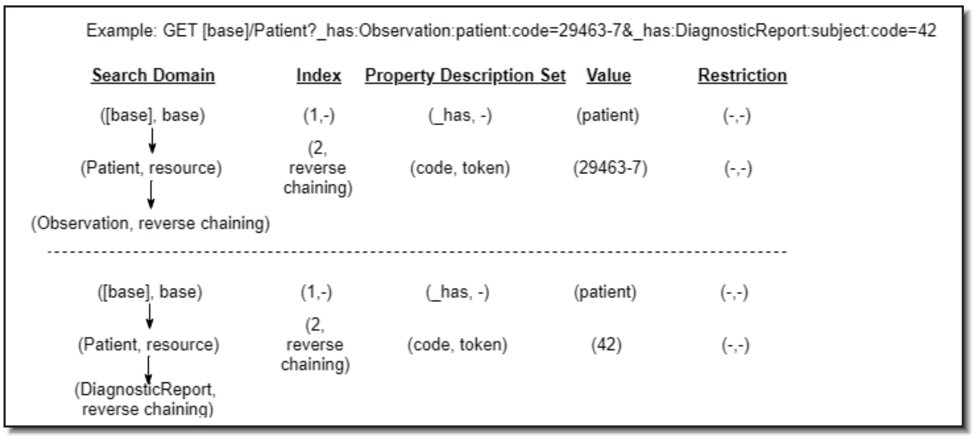
Figure 30: FHIR Components Model: Composite search



Example: GET [base]/Patient?_has:Observation:patient:code=29463-7

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,-)	(_has, -)	(patient)	(-,-)
↓				
(Patient, resource)	(2, reverse chaining)	(code, token)	(29463-7)	(-,-)
↓				
(Observation, reverse chaining)				

Figure 31: FHIR Components Model: Reverse chaining



Example: GET [base]/Patient?_has:Observation:patient:code=29463-7&_has:DiagnosticReport:subject:code=42

Search Domain	Index	Property Description Set	Value	Restriction
([base], base)	(1,-)	(_has, -)	(patient)	(-,-)
↓				
(Patient, resource)	(2, reverse chaining)	(code, token)	(29463-7)	(-,-)
↓				
(Observation, reverse chaining)				
([base], base)	(1,-)	(_has, -)	(patient)	(-,-)
↓				
(Patient, resource)	(2, reverse chaining)	(code, token)	(42)	(-,-)
↓				
(DiagnosticReport, reverse chaining)				

Figure 32: FHIR Components Model:
Reverse chaining with independently processed parameters

F. Appendix: Change Requests FHIR Standard

All of the following change requests have been filled in the FHIR Change Request Tracker Gforge [126] in the context of this thesis and are documented for future reference:

Non-substantive Change Requests:

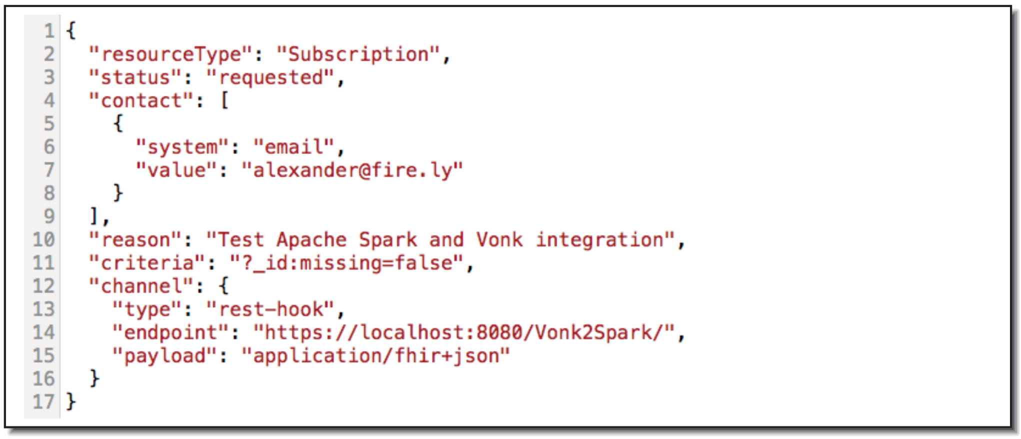
- Gforge #16016 - String search should be insensitive to all combining characters
- Gforge #16017 - Specify the behaviour if canonically equivalent characters are used in a string search
- Gforge #17219 - _filter example is invalid
- Gforge #17220 - Allow multiple :modifiers on a single search parameter?
- Gforge #17233 - state-on-date search parameter does not exist
- Gforge #17236 - Link to Forge is broken
- Gforge #17240 - ContactPoint links to the wrong data type
- Gforge #17436 - Encounter length is not a valid example for number search
- Gforge #17461 - Advanced text handling uses _text instead of text
- Gforge #17489 - _filter does not allow a paramName starting with "_"

Substantive Change Requests:

- Gforge #17488 - _filter is not formally defined in a SearchParameter resource

Any details about the change request can be looked up by the corresponding ID. Due to a still on-going discussion process - at the time of writing - no (preliminary) results are included in this thesis.

G. Appendix: FHIR Subscription for forwarding all new resources



```
{
  "resourceType": "Subscription",
  "status": "requested",
  "contact": [
    {
      "system": "email",
      "value": "alexander@fire.ly"
    }
  ],
  "reason": "Test Apache Spark and Vonk integration",
  "criteria": "?_id:missing=false",
  "channel": {
    "type": "rest-hook",
    "endpoint": "https://localhost:8080/Vonk2Spark/",
    "payload": "application/fhir+json"
  }
}
```

Figure 33: FHIR Subscription for forwarding all new resources

List of Figures

List of Tables

References

[1] HL7.org. (2017). Overview - FHIR v3.0.1, [Online]. Available: https://hl7.org/fhir/STU3/index.html (visited on 2018-02-06).

[2] J. C. Mandel *et al.*, “SMART on FHIR: a standards-based, interoperable apps platform for electronic health records.”, *Journal of the American Medical Informatics Association*, 2016-02, ISSN: 1527-974X. DOI: 10.1093/jamia/ocv189.

[3] D. F. Sittig and A. Wright, “What makes an EHR “open” or interoperable?”, *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 1099–1101, 2015. DOI: 10.1093/jamia/ocv060.

[4] F. C. Bourgeois *et al.*, “Patients treated at multiple acute health care facilities: Quantifying information fragmentation”, *Archives of Internal Medicine*, vol. 170, no. 22, pp. 1994–1995, 2010. DOI: 10.1001/archinternmed.2010.439.

[5] R. M. Califf *et al.*, “Transforming Evidence Generation to Support Health and Health Care Decisions”, *New England Journal of Medicine*, vol. 375, no. 24, pp. 2395–2400, 2016. DOI: 10.1056/NEJMsb1610128.

[6] K. B. Wagholikar *et al.*, “Evolving Research Data Sharing Networks to Clinical App Sharing Networks”, *AMIA Jt Summits Transl Sci Proc*, vol. 2017, pp. 302–307, 2017-07, 2613028[PII], ISSN: 2153-4063.

[7] HL7 FHIR Foundation. (2018). Implementation Registry | HL7 FHIR Foundation, [Online]. Available: http://www.fhir.org/implementations/registry (visited on 2018-01-27).

[8] Health Level Seven International, *The Argonaut Project: Project Charter*, 2014.

[9] JASON, *Data for Individual Health*, 2014.

[10] E. Bozdag, “Data Portability Under GDPR: Technical Challenges”, *SSRN Electronic Journal*, 2018. DOI: 10.2139/ssrn.3111866.

[11] R. A. Bloomfield *et al.*, “Opening the Duke electronic health record to apps: Implementing SMART on FHIR”, *International Journal of Medical Informatics*, vol. 99, pp. 1–10, 2017-03. DOI: 10.1016/j.ijmedinf.2016.12.005.

[12] S. N. Kasthurirathne *et al.*, “Towards Standardized Patient Data Exchange: Integrating a FHIR Based API for the Open Medical Record System”, *Studies in Health Technology and Informatics*, vol. 216, no. MEDINFO 2015: eHealth-enabled Health, pp. 932–932, 2015. DOI: 10.3233/978-1-61499-564-7-932.

[13] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential”, *Health Information Science and Systems*, vol. 2, no. 1, pp. 3–4, 2016-02-07. DOI: 10.1186/2047-2501-2-3.

[14] HL7.org. (2017). Search - FHIR v3.0.1, [Online]. Available: https://www.hl7.org/fhir/STU3/search.html (visited on 2018-02-12).

[15] J. C. Anderson *et al.*, *CouchDB: The Definitive Guide Time to Relax*, 1st. O’Reilly Media, Inc., 2010, pp. 53ff. ISBN: 0596155891, 9780596155896.

[16] I. Basho Technologies. (2017). Advanced MapReduce, [Online]. Available: http://docs.basho.com/riak/kv/2.2.3/developing/app-guide/advanced-mapreduce/ (visited on 2018-02-13).

[17] J. Robie *et al.*, *JSONiq: XQuery for JSON*, 2011.

[18] K. S. Beyer *et al.*, "Jaql: A Scripting Language for Large Scale Semistructured Data Analysis.", vol. 4, pp. 1272–1283, 2011-08.

[19] M. Bach and A. Werner, "Standardization of nosql database languages", in *Beyond Databases, Architectures, and Structures*, S. Kozielski *et al.*, Eds., Cham: Springer International Publishing, 2014, pp. 50–60, ISBN: 978-3-319-06932-6.

[20] Health Level Seven International. (2016). Fundamental Principles of FHIR, [Online]. Available: http://wiki.hl7.org/index.php?title=Fundamental_Principles_of_FHIR (visited on 2018-02-22).

[21] M. Smith *et al.*, *Best Care at Lower Cost: The Path to Continuously Learning Health Care in America*. The National Academies Press, 2012, ISBN: 978-0-309-26073-2.

[22] D. W. Bates, "The quality case for information technology in healthcare", *BMC Medical Informatics and Decision Making*, vol. 2, no. 1, 2002. DOI: 10.1186/1472-6947-2-7.

[23] Alliance for Health Reform, *High and Rising Costs of Health Care in the U.S. - The Challenge: Changing the Trajectory*, 2012. [Online]. Available: http://www.allhealthpolicy.org/wp-content/uploads/2017/03/Alliance_for_Health_Reform_121.pdf (visited on 2018-03-07).

[24] D. A. Conrad, "The Theory of Value-Based Payment Incentives and Their Application to Health Care", *Health Services Research*, vol. 50, pp. 2057–2089, 2015. DOI: 10.1111/1475-6773.12408.

[25] H. Wilson *et al.*, "Value based Healthcare", *Advances in Management*, vol. 9, no. 1, pp. 1–8, 2016.

[26] Rural Health IT. (2016). How Value-Based Healthcare depends on Interoperability, [Online]. Available: https://ruralhealthit.com/blog/2016/1/11/how-value-based-healthcare-depends-on-interoperability (visited on 2018-03-07).

[27] D. Shaller, *PATIENT-CENTERED CARE: WHAT DOES IT TAKE?*, 2007. [Online]. Available: http://www.commonwealthfund.org/~/media/Files/Publications/Fund%20Report/2007/Oct/Patient%20Centered%20Care%20%20What%20Does%20It%20Take/Shaller_patient%20centeredcarewhatdoesittake_1067%20pdf.pdf (visited on 2018-03-07).

[28] Ontario Medical Association, *Policy Paper - Patient Centered Care*, 2010. [Online]. Available: http://hsprn.ca/seminarFiles/14id1335542366.pdf (visited on 2018-03-07).

[29] S. Qureshi *et al.*, "Mobile Access for Patient Centered Care: The Challenges of Activating Knowledge through Health Information Technology", in *2015 48th Hawaii International Conference on System Sciences*, IEEE, 2015. DOI: 10.1109/hicss.2015.389.

[30] M. Marsilio *et al.*, “Health care multidisciplinary teams”, *Health Care Management Review*, vol. 42, no. 4, pp. 303–314, 2017. DOI: 10.1097/hmr.0000000000000115.

[31] J. Kaipio *et al.*, “Usability problems do not heal by themselves: National survey on physicians’ experiences with EHRs in Finland”, *International Journal of Medical Informatics*, vol. 97, pp. 266–281, 2017. DOI: 10.1016/j.ijmedinf.2016.10.010.

[32] “Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models”, International Organization for Standardization, Standard, 2011.

[33] B. B. Page, “Exploring Organizational Culture for Information Security in Healthcare Organizations: A Literature Review”, in *2017 Portland International Conference on Management of Engineering and Technology (PICMET)*, IEEE, 2017. DOI: 10.23919/picmet.2017.8125471.

[34] S. Bowman, “Impact of Electronic Health Record Systems on Information Integrity: Quality and Safety Implications”, *Perspect Health Inf Management*, vol. 10, no. Fall, p. 1c, 2013, ISSN: 1559-4122.

[35] K. Kosanke, “ISO Standards for Interoperability: a Comparison”, in *Interoperability of Enterprise Software and Applications*, Springer-Verlag, pp. 55–64. DOI: 10.1007/1-84628-152-0_6.

[36] H. Kubicek *et al.*, “Layers of Interoperability”, in *Organizational Interoperability in E-Government*, Springer Berlin Heidelberg, 2011, pp. 85–96. DOI: 10.1007/978-3-642-22502-4_7.

[37] D. Broyles *et al.*, “The Evolving Health Information Infrastructure”, in *Health Information Exchange*, Elsevier, 2016, pp. 107–122. DOI: 10.1016/b978-0-12-803135-3.00007-4.

[38] M. Hosseini and B. E. Dixon, “Syntactic Interoperability and the Role of Standards”, in *Health Information Exchange*, Elsevier, 2016, pp. 123–136. DOI: 10.1016/b978-0-12-803135-3.00008-6.

[39] T. Benson and G. Grieve, *Principles of Health Interoperability*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-30370-3.

[40] H. van der Veer and A. Wiles, *ETSI White Paper No. 3 - Achieving Technical Interoperability - the ETSI Approach*, 2008.

[41] HL7 International, *HL7 Strategic Plan – 2017*, 2017. [Online]. Available: http://www.hl7.org/documentcenter/public/twitter/HL7_TWITTER_20171221.pdf (visited on 2018-03-16).

[42] openEHR Foundation. (2018). Welcome to openEHR, [Online]. Available: http://www.openehr.org/home (visited on 2018-04-17).

[43] E. Haas, *Introduction to FHIR*, Online Tutorial, 2015. [Online]. Available: https://gforge.hl7.org/gf/project/fhir/scmsvn/?action=browse&path=%2F%2Acheckout%2A%2Ftags%2F2016May%2Fpresentations%2F2016-01%2520Tutorials%2FIntroduction%2520to%2520FHIR-eh.pptx.

[44] R. Spronk. (2014). The early history of health Level 7, [Online]. Available: http://www.ringholm.com/docs/the_early_history_of_health_level_7_HL7.htm (visited on 2018-03-19).

Technology
Arts Sciences
TH Köln

[45] D. Bender and K. Sartipi, “HL7 FHIR: An Agile and RESTful approach to healthcare information exchange”, in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, IEEE, 2013. DOI: 10.1109/cbms.2013.6627810.

[46] C. N. Mead, “Data Interchange Standards in Healthcare IT - Computable Semantic Interoperability: Now Possible but Still Difficult, Do We Really Need a Better Mousetrap?”, *Journal of Healthcare Information Management*, vol. 20, no. 1, 2006.

[47] A.-M. Shakir. (2014). Hl7 reference information model, [Online]. Available: https://www.slideshare.net/AShakir/hl7-reference-information-model (visited on 2018-03-20).

[48] B. Smith and W. Ceusters, “HL7 RIM: an incoherent standard”, vol. 124, pp. 133–8, 2006-02.

[49] G. Schadow *et al.*, “The HL7 Reference Information Model Under Scrutiny”, vol. 124, pp. 151–6, 2006-02.

[50] R. Spronk. (2011). RFH (Resources for Health): HL7 version 3 taken to the next step, [Online]. Available: http://www.ringholm.com/column/RFH_HL7_version_3_taken_to_the_next_step.htm (visited on 2018-04-18).

[51] G. Grieve *et al.* (2012). Introduction to HL7 FHIR, [Online]. Available: https://gforge.hl7.org/gf/project/fhir/scmsvn/?action=browse&path=%2F%2Acheckout%2A%2Fdstu2-last%2Fpresentations%2FIntroduction%2520to%2520FHIR.pptx (visited on 2018-03-21).

[52] HL7.org. (2017). 5.2 Resource CapabilityStatement - Content, [Online]. Available: http://hl7.org/fhir/STU3/capabilitystatement.html (visited on 2018-03-22).

[53] H. van der Linden *et al.*, “Inter-organizational future proof EHR systems”, *International Journal of Medical Informatics*, vol. 78, no. 3, pp. 141–160, 2009. DOI: 10.1016/j.ijmedinf.2008.06.013.

[54] HL7.org. (2017). 2.21.0 RESTful API, [Online]. Available: https://www.hl7.org/fhir/STU3/http.html (visited on 2018-03-23).

[55] HL7.org. (2017). 2.24 Messaging using FHIR Resources, [Online]. Available: https://www.hl7.org/fhir/STU3/messaging.html (visited on 2018-03-24).

[56] R. Spronk. (2017). Messaging - The unloved paradigm, [Online]. Available: https://www.slideshare.net/DevDays/20171127-rene-spronkmessagingtheunlovedparadigm (visited on 2018-03-24).

[57] R. H. Dolin *et al.*, “The HL7 Clinical Document Architecture”, *Journal of the American Medical Informatics Association*, vol. 8, no. 6, pp. 552–569, 2001. DOI: 10.1136/jamia.2001.0080552.

[58] HL7.org. (2017). 2.41 Resource Composition - Content, [Online]. Available: https://www.hl7.org/fhir/STU3/composition.html (visited on 2018-03-26).

[59] Intel Corporation, *SOA in Healthcare A Conceptual Overview*, White Paper, 2009. [Online]. Available: http://www.itoamerica.com/media/pdf/intel/soa_in_healthcare.pdf.

[60] MuleSoft, Inc. (n.d.). Service Orchestration and SOA, [Online]. Available: https://www.mulesoft.com/de/resources/esb/service-orchestration-and-soa (visited on 2018-03-25).

[61] HL7.org. (2017). 2.25 Using Resources with Services and Service-oriented Architecture, [Online]. Available: https://www.hl7.org/fhir/STU3/services.html (visited on 2018-03-25).

[62] HL7.org. (2017). 2.29 DomainResource Resource, [Online]. Available: https://www.hl7.org/fhir/STU3/domainresource.html (visited on 2018-03-26).

[63] SmartConData GmbH. (2018). HL7 FHIR, [Online]. Available: https://www.smartcondata.de/start/know-how/fhir/l (visited on 2018-05-16).

[64] HL7.org. (2017). 5.3 Resource StructureDefinition - Content, [Online]. Available: http://hl7.org/fhir/STU3/structuredefinition.html (visited on 2018-03-26).

[65] HL7.org. (2017). 2.3.0 Resource References, [Online]. Available: https://www.hl7.org/fhir/STU3/references.html (visited on 2018-04-04).

[66] SNOMED International. (2018). SNOMED CT - The Global Language of Healthcare, [Online]. Available: https://www.snomed.org/snomed-ct (visited on 2018-04-03).

[67] Regenstrief Institute, Inc. (2017). LOINC - The international standard for identifying health measurements, observations, and documents., [Online]. Available: https://loinc.org/ (visited on 2018-04-03).

[68] HL7.org. (2017). 4.0 Terminology Module, [Online]. Available: https://www.hl7.org/fhir/STU3/terminology-module.html (visited on 2018-03-28).

[69] HL7.org. (2017). 5.1.0 Profiling FHIR, [Online]. Available: https://www.hl7.org/fhir/STU3/profiling.html (visited on 2018-04-06).

[70] HL7 FHIR Foundation. (2018). Implementation Registry | HL7 FHIR Foundation, [Online]. Available: http://www.fhir.org/guides/registry (visited on 2018-07-31).

[71] Firely. (2017). Simplifier.net - Home, [Online]. Available: https://simplifier.net/ (visited on 2018-04-09).

[72] Firely. (n.d.). Get started with Simplifier, [Online]. Available: https://simplifier.net/guide/ProfilingAcademy/GetstartedwithSimplifier (visited on 2018-04-09).

[73] "Health informatics – Electronic health record – Definition, scope and context", International Organization for Standardization, Standard, 2005.

[74] "Health informatics – Electronic health record communication – Part 1: Reference model", International Organization for Standardization, Standard, 2008.

[75] D. Garets and M. Davis, "Electronic Medical Records vs. Electronic Health Records: Yes, There Is a Difference", 2006.

[76] Apple Inc. (2018). HealthKit, [Online]. Available: https://developer.apple.com/documentation/healthkit (visited on 2018-06-07).

[77] Health Level Seven Inc., "HL7 EHR System Functional Model: A Major Development Towards Consensus on Electronic Health Record System Functionality", 2004.

[78] Y. Wang *et al.*, “Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations”, *Technological Forecasting and Social Change*, vol. 126, pp. 3–13, 2018. DOI: 10.1016/j.techfore.2015.12.019.

[79] B. Mishra and R. Kumar, *Big Data Management and the Internet of Things for Improved Health Systems*, ser. Advances in Healthcare Information Systems and Administration. IGI Global, 2018, pp. 232, ISBN: 9781522552239.

[80] A. Rajkomar *et al.*, “Scalable and accurate deep learning with electronic health records”, *npj Digital Medicine*, vol. 1, no. 1, 2018. [Online]. Available: https://doi.org/10.1038/s41746-018-0029-1.

[81] C. Bhatt *et al.*, Eds., *Internet of Things and Big Data Technologies for Next Generation Healthcare*. Springer International Publishing, 2017, pp. 137ff. DOI: 10.1007/978-3-319-49736-5.

[82] Google LLC. (2018). Making Healthcare Data Work Better with Machine Learning, [Online]. Available: https://ai.googleblog.com/2018/03/making-healthcare-data-work-better-with.html (visited on 2018-06-11).

[83] A. Gandomi and M. Haider, “Beyond the hype: Big data concepts, methods, and analytics”, *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015. DOI: 10.1016/j.ijinfomgt.2014.10.007.

[84] Nictiz. (2018). HCIM Mainpage, [Online]. Available: https://zibs.nl/wiki/HCIM_Mainpage (visited on 2018-06-27).

[85] Nictiz. (2017). MedMij FHIR use case BgZ , version 2017.01, [Online]. Available: https://informatiestandaarden.nictiz.nl/wiki/MedMij:V2017.01_FHIR_BGZ (visited on 2018-06-28).

[86] S. Wang and K.-L. Zhang, “Searching Databases with Keywords”, *Journal of Computer Science and Technology*, vol. 20, no. 1, pp. 55–62, 2005-01. DOI: 10.1007/s11390-005-0006-4.

[87] P. Atzeni *et al.*, *Database Systems - Concepts, Languages and Architectures*. The McGraw-Hill Companies, 1999-01, ISBN: 0-07-709500-6.

[88] T. J. Bothma, “Filtering and adapting data and information in the online environment in response to user needs”, *e-Lexicography: The Internet, Digital Initiatives and Lexicography*, pp. 71–102, 2011.

[89] C. Lanquillon, “Enhancing Text Classification to Improve Information Filtering”, PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2001.

[90] HL7.org. (2017). 2.19 Outstanding Issues, [Online]. Available: https://hl7.org/fhir/STU3/todo.html (visited on 2018-04-10).

[91] HL7.org. (2017). 2.21.1 Search, [Online]. Available: http://hl7.org/fhir/STU3/search.html (visited on 2018-04-11).

[92] W. Kim *et al.*, “Semantic Web Based Intelligent Product and Service Search Framework for Location-Based Services”, in *Computational Science and Its Applications – ICCSA 2005*, Springer Berlin Heidelberg, 2005, pp. 103–112. DOI: 10.1007/11424925_13.

[93] HL7.org. (2017). 5.6.1 Scope and Usage, [Online]. Available: https://www.hl7.org/fhir/STU3/compartmentdefinition.html (visited on 2018-04-16).

[94] HL7.org. (2018). FHIRPath (STU1 Release) - Navigation model, [Online]. Available: http://hl7.org/fhirpath/#navigation-model (visited on 2018-04-16).

[95] HL7.org. (2017). FHIR Conformance QA Criteria, [Online]. Available: http://wiki.hl7.org/index.php?title=FHIR_Conformance_QA_Criteria (visited on 2018-04-16).

[96] HL7.org. (2017). 4.2.12.12 Code System http://hl7.org/fhir/search-param-type, [Online]. Available: http://hl7.org/fhir/stu3/codesystem-search-param-type.html (visited on 2018-04-20).

[97] HL7.org. (2017). 2.26.0 Data Types, [Online]. Available: http://hl7.org/fhir/stu3/datatypes.html (visited on 2018-04-23).

[98] M. A. Raebel *et al.*, "Electronic clinical laboratory test results data tables: lessons from Mini-Sentinel", *Pharmacoepidemiology and Drug Safety*, vol. 23, no. 6, pp. 609–618, 2014. [Online]. Available: https://doi.org/10.1002/pds.3580.

[99] Regenstrief Institute, Inc. and the UCUM Organization. (2017). THE UNIFIED CODE FOR UNITS OF MEASURE, [Online]. Available: http://unitsofmeasure.org/ucum.html (visited on 2018-05-21).

[100] HL7.org. (2017). Extension: subsumes, [Online]. Available: https://www.hl7.org/fhir/extension-codesystem-subsumes.html (visited on 2018-05-22).

[101] Unicode, Inc. (2017). Unicode Standard Annex #41 Common References for Unicode Standard Annexes, [Online]. Available: http://unicode.org/reports/tr29/#Grapheme_Cluster_Boundaries (visited on 2018-04-26).

[102] Unicode, Inc. (2017). Unicode Standard Annex #15 Unicode Normalization Forms, [Online]. Available: http://www.unicode.org/reports/tr15/#Canon_Compat_Equivalence (visited on 2018-04-26).

[103] HL7.org. (2017). 2.21.2 _filter Parameter, [Online]. Available: http://hl7.org/fhir/stu3/search_filter.html (visited on 2018-05-29).

[104] S. T. Rosenbloom *et al.*, "Data from clinical notes: a perspective on the tension between structure and flexible documentation", *Journal of the American Medical Informatics Association*, vol. 18, no. 2, pp. 181–186, 2011. [Online]. Available: https://doi.org/10.1136/jamia.2010.007237.

[105] A. Mathioudakis *et al.*, "How to keep good clinical records", *Breathe*, vol. 12, no. 4, pp. 369–373, 2016. [Online]. Available: https://doi.org/10.1183/20734735.018016.

[106] G. Gonzalez-Hernandez *et al.*, "Capturing the Patient's Perspective: a Review of Advances in Natural Language Processing of Health-Related Text", *Yearbook of Medical Informatics*, vol. 26, no. 01, pp. 214–227, 2017. [Online]. Available: https://doi.org/10.15265/iy-2017-029.

[107] W. Sun *et al.*, "Temporal reasoning over clinical text: the state of the art", *Journal of the American Medical Informatics Association*, vol. 20, no. 5, pp. 814–819, 2013. [Online]. Available: https://doi.org/10.1136/amiajnl-2013-001760.

[108] H. Wu *et al.*, “SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research”, *Journal of the American Medical Informatics Association*, vol. 25, no. 5, pp. 530–537, 2018. [Online]. Available: https://doi.org/10.1093/jamia/ocx160.

[109] HL7.org. (2017). Getting Started with FHIR, [Online]. Available: http://hl7.org/fhir/modules.html (visited on 2018-05-25).

[110] HL7.org. (2014). 0 Welcome to FHIR, [Online]. Available: http://hl7.org/fhir/DSTU1/index.html (visited on 2018-05-25).

[111] “Quality management systems — Fundamentals and vocabulary”, International Organization for Standardization, Standard, 2015.

[112] Health Level Seven International. (2018). HL7 Fast Healthcare Interoperability Resources Specification (aka FHIR®), Release 3 (STU), [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=449 (visited on 2018-06-11).

[113] D. O. Case, *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior (Library and Information Science)*. Emerald Group Publishing Limited, 2007, ISBN: 0123694302.

[114] HL7.org. (2017). 5.6.1 Scope and Usage, [Online]. Available: https://www.hl7.org/fhir/stu3/compartmentdefinition.html#scope (visited on 2018-06-18).

[115] G. Grieve. (2018). Reference Implementation Server for the FHIR Specification, [Online]. Available: https://github.com/grahamegrieve/fhirserver (visited on 2018-06-29).

[116] O. Andrei and H. Kirchner, “A Rewriting Calculus for Multigraphs with Ports”, *Electronic Notes in Theoretical Computer Science*, vol. 219, pp. 67–82, 2008. DOI: 10.1016/j.entcs.2008.10.035.

[117] P. T. Wood, “Query languages for graph databases”, *ACM SIGMOD Record*, vol. 41, no. 1, pp. 50–60, 2012. DOI: 10.1145/2206869.2206879.

[118] P. B. Baeza, “Querying graph databases”, in *Proceedings of the 32nd symposium on Principles of database*, ACM Press, 2013. DOI: 10.1145/2463664.2465216.

[119] HL7.org. (2017). 5.6.1 Scope and Usage, [Online]. Available: http://hl7.org/fhir/stu3/searchparameter-registry.html (visited on 2018-07-17).

[120] P. Barceló *et al.*, “Querying graph patterns”, in *Proceedings of the 30th symposium on Principles of database systems of data - PODS*, ACM Press, 2011. DOI: 10.1145/1989284.1989307.

[121] L. Libkin and D. Vrgoč, “Regular path queries on graphs with data”, in *Proceedings of the 15th International Conference on Database Theory - ICDT ’12*, ACM Press, 2012. DOI: 10.1145/2274576.2274585.

[122] Cerner. (2018). Bunsen: FHIR Data with Apache Spark, [Online]. Available: https://engineering.cerner.com/bunsen/ (visited on 2018-08-11).

[123] H. Karau *et al.*, *Learning Spark: Lightning-Fast Big Data Analysis*. O’Reilly Media, 2015, ISBN: 1449358624.

[124] University Health Network. (2018). HAPI FHIR - The Open Source FHIR API for Java, [Online]. Available: http://hapifhir.io (visited on 2018-08-11).

[125] HL7.org. (2017). 2.46 Resource Subscription - Content, [Online]. Available: https://www.hl7.org/fhir/stu3/subscription.html (visited on 2018-08-11).

[126] ——, (2018). FHIR Change Request Tracking System, [Online]. Available: https://gforge.hl7.org/gf/project/fhir/tracker/ (visited on 2018-08-03).