HL7® FHIR® -
Fast Healthcare Interoperability Resources
An Introduction

Frank Oemig[a] [1]

[a]*IT-Consulting in Healthcare, Mülheim, Germany*

Abstract

After more than 20 years of experience with HL7 v2.x and 15 years with Version 3.0, HL7 International raised the question how the experience and knowledge gained could be migrated to a new, but simpler standard that is also more modern. The result is a new framework called FHIR (Fast Healthcare Interoperability Resources) that is still under development and testing for more than 10 years now and will be explained in the following. The interest in FHIR is rising not only worldwide, but also in Germany leading to more FHIR-based specifications as the foundation for open and standardized interfaces.

Keywords: HL7, FHIR, interoperability, implementation guide, profile

Zusammenfassung

HL7 International (http://www.hl7.org) hatte nach 20 Jahren HL7 v2.x und 15 Jahren Version 3.0 überlegt, wie man die Erfahrungen aus den vergangenen Entwicklungen zu einem neuen Standard zusammenführen könnte, der einen Einsatz gleichzeitig einfacher macht. Herausgekommen ist ein umfangreiches Rahmenwerk, das sich FHIR (Fast Healthcare Interoperability Resources) nennt, seit mehr als 10 Jahren in der Entwicklung ist und im nachfolgenden Artikel ausführlicher vorgestellt wird. Weltweit und neuerdings auch in Deutschland findet FHIR immer stärkere Beachtung und wird deshalb zur Grundlage weiterer Ausarbeitungen und Vorgaben für offene Schnittstellen.

Schlüsselwörter: HL7, FHIR, Interoperabilität, Implementierungsleitfaden, Profile

1 Introduction

HL7 version 3.0 had not achieved a significant breakthrough in the market despite massive investments in the development of the basics, methodologies and tools until 2009. The fundamental and accompanying steps in development have formed a concrete understanding of how to deal with standards, how they need to be developed from the process model, how important an architectural foundation is, and that data types, information models and vocabulary must be separated. At the same time, it has become clear that proper compliance with all these mechanisms together has a complexity that cannot be managed in a simple way.

At the same time, various advances in the field of smartphones and associated apps took place, and a trend from SOAP to REST, from XML to JSON as well as various other parallel development steps in the field of communication technology were also evident.

That is why HL7 International asked what a standard "today", i.e. 2009, would have to look like if one could start again with the current knowledge on a green field. This led to the so-called "Fresh Look Initiative", which was to investigate this issue by agreeing with the relevant working groups at HL7 International and then bringing the results together. This taskforce then returned in 2011 with the "Resources for Healthcare", which in 2012 became the "Fast Healthcare Interoperability

[1] Dr. Frank Oemig, IT-Consulting in Healthcare, Mülheim, Germany, hl7@oemig.de

Resources" – abbreviated as "FHIR" and pronounced "fire" – with the intent to include the best of the existing standards, although there are still options for improvements. This new framework is designed to simplify and modernize interface specifications for use with different systems.

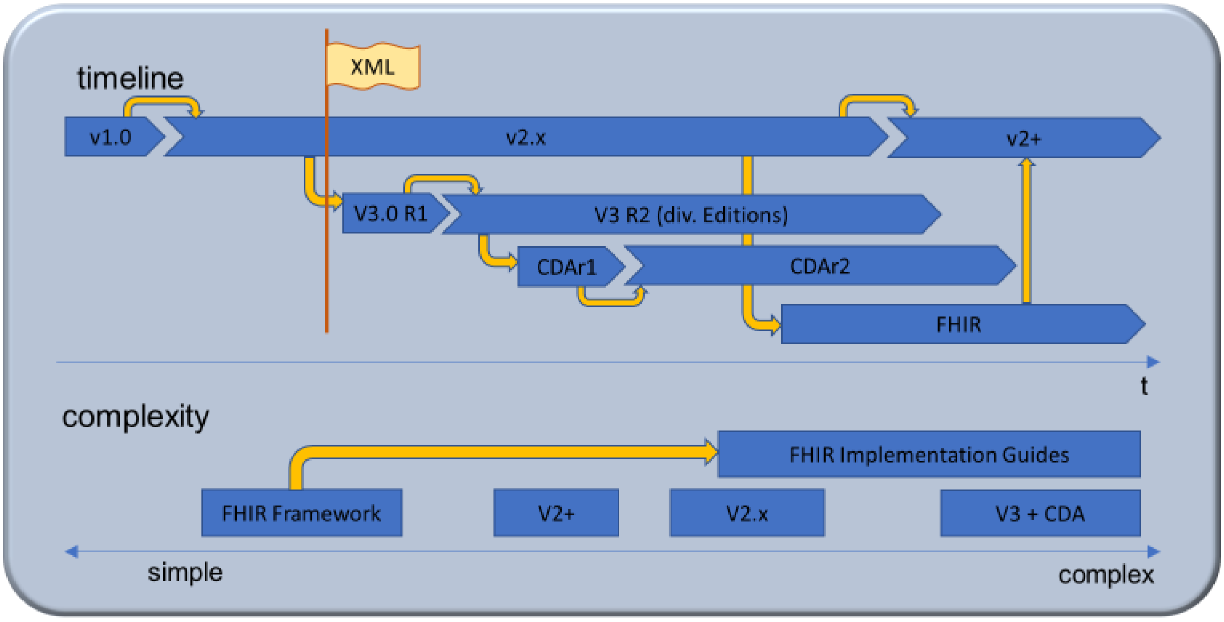

Fig. 1: Alignment of FHIR into the family of HL7 Standards.

Fig. 1 illustrates the history of standard developments both in chronological order and the classification of complexity. The FHIR framework itself is less complex and easier when it comes to merging various aspects such as data types, presentation of technical details, direct integration of technical implementation in the form of XML and JSON and the provision of many examples (see Fig. 3) than the other HL7 standards v2.x (v2+) and version 3.0 with CDA.

After the first version (draft) as DSTU 1 (2014) additional revisions were issued as further development. Since 2019, there has been a version (R4) that contains normative elements for the first time. Currently (2/2021) work is underway on R4B and R5, with a balloting planned for May 2021. For subsequent releases, more versions are expected to be added.

FHIR provides the underlying framework that is still being developed. To be compliant with FHIR, no change to the framework itself is allowed. Any additional details and requirements essential for implementation/realization are described in implementation guides, which are developed separately.

2 Foundation

As specified in the FHIR specification [1], the best and proven aspects of the HL7 standards v2.x, V3 and CDA are used. The most important aspect is component orientation ("building blocks") contained in FHIR in the form of "resources". In addition, there is the REST design principle for http-based transmission of data.

A basic problem in standard development is the completeness of the specification. HL7 version 2.9 currently contains nearly 2,500 data elements, but only a very small part of them is commonly used. Most data elements serve special cases that are relatively rarely implemented or used. FHIR wanted to address this in advance by defining only those data elements or attributes for which there is also a need in 80% of the cases used. This significantly reduces the discussion and specification effort and thus accelerates it. At the same time, a mechanism, which is later explained under the heading "extension", should prevent the untethered expansion, as happened with the HL7 v2.x Z-segments, for example. On the one hand, a logical opening was achieved, but it is still technically closed. In other words, FHIR can be extended to support use cases as needed, but unlike v2.x FHIR prescribes a rigorous mechanism for doing so. This allows implementers and systems to be able to read and understand the extension by the definition itself. Despite this "restriction", FHIR R4 currently contains about 150 blocks (resources) with a considerable number of attributes each, which are already well in the four-digit range.

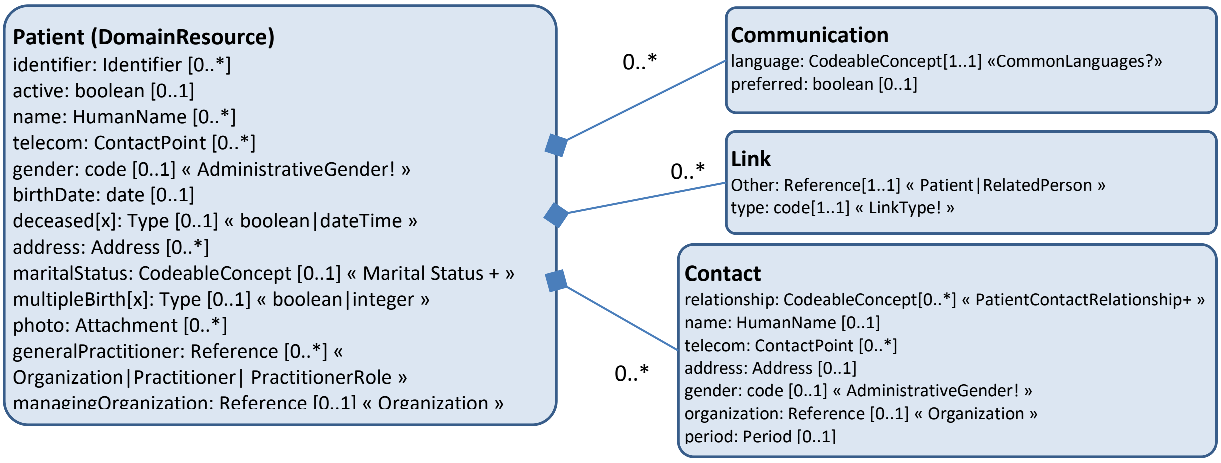

Fig. 2: FHIR Patient Resource [10]

For the specification, FHIR uses UML based mini models (Fig. 2) in addition to a hierarchical tree representation and XML, JSON and other variants (Turtle, ...). In the specification this is shown directly side by side, thus ensuring an easier understanding and thus facilitates higher acceptance.

Because there is no architectural framework behind FHIR that enforces compliance with certain conventions, the attribute names can be self-explanatorily named simplifying the hierarchic structures, resulting in easier applicability.

The primary building blocks in FHIR are resources that can be made available in two different encodings in parallel. One encoding is based on XML, the other on the Java Script Object Notation (JSON), and the two can be converted without any loss of relevant information. Fig. 3 demonstrates how the same information for a particular patient resource (Fig. 2) is presented in both encodings. The decision about what to use is made using the content-type attribute in the http header. A FHIR server should support both variants to simplify communication with the client while ensuring maximum flexibility.

XML	JSON
<Patient xmlns=”http://hl7.org/fhir”>	{ "resourceType": "Patient",
<id value=”cda”/>	"id": "xcda",
<text> <status value=”generated”/> <div xmlns=”http://www.w3.org/1999/xhtml”> <p>Henry LEVIN the 7th, DOB 24-Sept 1932</p> <p>MRN: 123456</p> </div> </text>	"text": { "status": "generated", "div": " Henry LEVIN the 7th, DOB 24-Sept 1932 MRN: 123456 " },

<extension url="http://hl7.org/fhir/
StructureDefinition/us-core-race">
<valueCodeableConcept>
<coding>
<system value=
"http://hl7.org/fhir/v3/Race"/>
<code value="1096-7"/>
</coding>
</valueCodeableConcept>
</extension>
	"extension": [
	{
	"url": "http://hl7.org/fhir/
	StructureDefinition/
	us-core-race",
	"valueCodeableConcept": {
	"coding": [{
	"system":
	"http://hl7.org/fhir/v3/Race",
	"code": "1096-7"

XML	JSON
	}] } },

<identifier>
<use value=”usual”/>
<type>
<coding>
<system value=
”http://hl7.org/fhir/v2/0203”/>
<code value=”MR”/>
</coding>
</type>
<system value=
”urn:oid:2.16.840.1.113883.19.5”/>
<value value=”12345MR”/>
</identifier>
	"identifier": [
	{
	"use": "usual",
	"type": {
	"coding": [{
	"system":
	"http://hl7.org/fhir/v2/0203",
	"code": "MR"
	}]
	},
	"system":
	"urn:oid:2.16.840.1.113883.19.5",
	"value": "12345"
	}
	],

<active value=”true”/>	"active": true,

<name>
 	
<family value=”Levin”/>
<given value=”Henry”/>
<suffix value=”The 7th”/>
</name>
<gender value=”male”/>
<birthDate value=”1932-09-24”/>
	"name": [
	{
	"family": ["Levin"],
	"given": ["Henry"]
	}
	],
	"gender": "male",
	"birthDate": "1932-09-24",

<careProvider>
<type value=”Organization/1”/>
<display value=”Good Health Clinic”/>
</careProvider>
	"careProvider": {
	"reference": "Organization/
	2.16.840.1.113883.19.5",
	"display": "Good Health Clinic"
	}

</Patient>	}

Fig. 3: Example Instance of a FHIR Patient Resource in XML and JSON [11], [12]

The data elements are distributed among many different resources, which are roughly based on the classes of HL7 version 3 and are organized into categories to specify different services.

The resources cover very different categories that emphasize the integrative character of FHIR:

- administrative: Patient, Encounter, Location, Organization ...
- clinical: Observation, Concern, Allergy Intolerance, Procedure, ...
- devices: Devices, Metric, ...
- scheduling: Appointment, Slot, ...
- financial: Claim, Account, Coverage, ..
- research: Research Study, Research Definition, ..
- forms: Questionnaire, Questionnaire Response, ..
- technical: Structure Definition, Capability Statement, Endpoint, Test Script, ..
- vocabulary: Codesystem, Value Set, ..

2.1 Element Hierarchy

In FHIR, all artifacts can be derived from a common abstract root element as shown in Fig. 4. (Fig. 4 shows only a snippet with a few examples to illustrate the hierarchy.) From this base element, two different hierarchies are derived that make up the actual fascination of FHIR as illustrated on the left- or right-hand side of the figure. The base element itself provides the capabilities needed to build the resource hierarchy, such as extensions and metadata, thus enabling all derived artifacts to have the same basic properties. At the same time, this class hierarchy ensures that the element hierarchy is technically closed and that all the details are known. With this approach, even the extensions are not extensions but are a predefined mechanism for very specific constraints, which then become profile components [2].

The element hierarchy includes data types, which are shown with some exemplary specializations on the left-hand side of Fig. 4. Complex and primitive data types are used, some of which are based on the data types of HL7 v2.x and V3, such as:

- Simple/primitive, consisting of a single value
- Complex, but generally usable for names or addresses
- For metadata
- For specific purposes

The resources that fulfill the primary healthcare use cases and are used for implementation guides, such as Patient and Encounter, are specializations of an abstract domain resource, which in turn is based on an abstract resource specification. The latter ensures that each resource can contain metadata in addition to an identification for the instance. The referencing mechanism for resources (see Fig. 4) and an optional textual description of the resource are realized through the domain resources. The latter is also the anchor for extensions as well as the so-called modifier extensions which express those instances changed the original semantics and therefore require special treatment.

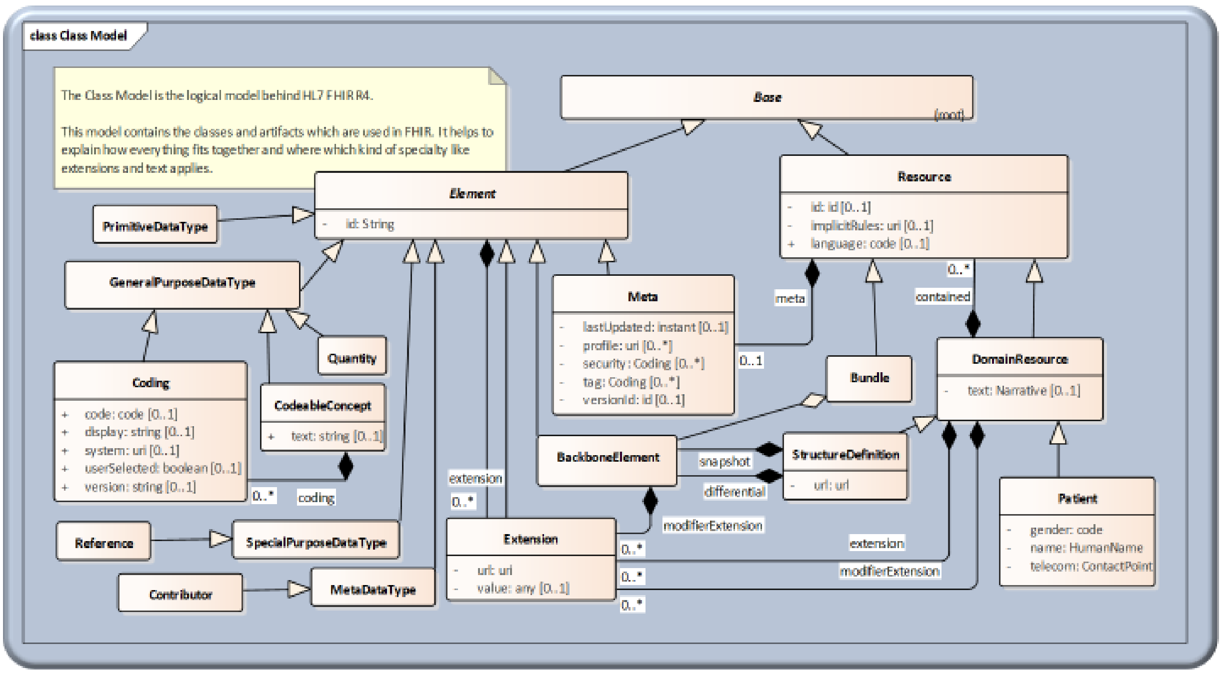

Fig. 4: HL7 FHIR Element Hierarchy (derived from the Specification [2])

Attributes of a resource are used to link to instances of other resources by references, or they are the foundation for additional attributes by using generic backbone elements, or they are defined directly by a specific data type. Within FHIR, some attributes allow for different types of data. For example, the attribute “deceased” in the patient resource can be instantiated as a boolean or date/time data type to express either the fact "died" or the exact date of death. (In this case, the information is accessed using either the deceasedBoolean or deceasedDateTime attribute.) Such optionality is then a predestined candidate for further restrictions in derived profiles or implementation guides.

It should be noted that empty attributes ("") are not allowed in an instance, so the cardinality refers directly to the occurrence with certain values. In addition, nearly all attributes in the FHIR framework are optional, i.e. there are no basic constraints, so the minimum cardinality is always "0" except for special resources.

2.2 Basic Paradigms

FHIR supports various communication paradigms [3]:

The *Composition Resource* provides documents using sections. These sections are nested as well as repeatable. In addition to text, they can also refer to other entries via references. Unlike CDA, this creates an equivalent list of instances that reference each other instead of a structured hierarchy.

Similarly, the message paradigm is also provided as messaging. The resource responsible for this is the *MessageHeader*, which provides information about who sends which message to whom and why. A *MessageDefinition*, which the MessageHeader can reference, allows for refining the message content.

For both, documents and message, all details are then also summarized and delivered via a *Bundle* Resource.

The RESTful API, which stands for Representational State Transfer and uses stateless operations to implement the Create, Read, Update, and Delete (CRUD) services, is the basis for FHIR [3]. FHIR's resources are the syntactic basis for packaging data and transmitting it via http request.

On the one hand, the FHIR framework offers standardized basic services, and on the other hand, it provides a way to define and offer its own services through *Operation Definition*, through which concrete actions can be carried out and thus, for example, the expansion of a value set from a definition or a drug interaction check can be realized. This eliminates a weakness that has been criticized by experts for years.

FHIR can not only be used as an API for conventional applications such as hospital information system (HIS) or general practitioner (GP) systems, FHIR instances can also be stored "as is" on so-called FHIR servers. With the latter, all information is then preserved, while in a facade only the details that are specifically supported by the server ("must support", see below) are used.

2.3 Linking via References

FHIR resources are the building blocks for developing more complex structures. The standard is relatively silent about exactly how presenting complex issues should be done. FHIR facilitates a fundamental property of web technology that allows an instance of a resource to be related to another instance by a reference.

Therefore, almost all resources contain some attributes that are declared as references. Fig. 5 illustrates as an example the expression and use of some references that are defined with the *Procedure Resource*.

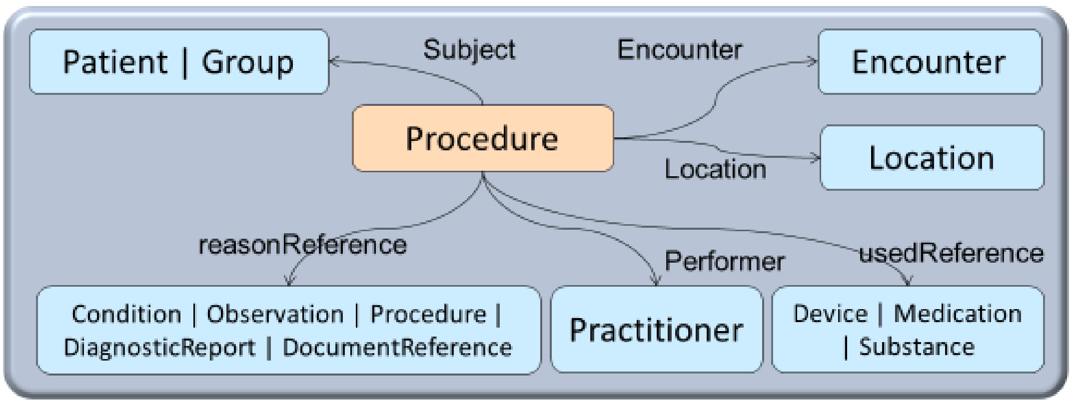

Fig. 5: References [13]

Each reference can consist of a URL and a text description for display. Fig. 6 shows an example of how such a reference in XML can then be expressed in concrete terms:

```xml
<performer>
  <actor>
    <reference value="Practitioner/example"/>
    <display value="Dr Cecil Surgeon"/>
  </actor>
</performer>
```

Fig. 6: Example Reference [16]

The use of references is very flexible: references can point as an absolute reference to resources anywhere on the internet, or relative to the resource from which it is referenced (within the same server), or internally (in a special compilation). Therefore, a network of resources can be built up easily and can be spread across multiple organizations. For example, one system might act as provider of administrative data while another might provide findings.

Instead of functioning as a reference, a resource could be directly embedded (contained) if it is not needed as a standalone instance. This further reduces the necessary administrative burden.

2.4 Bundling

The primary building block in FHIR is a resource. But transmitting or using just a single resource is only sufficient for simple contexts. In the simplest case, each resource can be used alone, which leads to the need for extensive communication, as multiple resources of the same or another type are always affected for specific use cases. Therefore, you typically need to combine multiple resources to create a specific context:

- Compilation of multiple actions for resources (transaction)
- Compilation of multiple query results in a list
- Combination of details, e.g. all values of a laboratory finding
- Building a history
- Detailed results on a document in a structured form
- And others

Such resources are thus grouped together by a special resource called a *Bundle*. The Bundle allows for simple data compilations to be expressed as a simple list or for more complex transactions.

2.5 http-Requests

The primary data exchange between communication partners takes place via standard HTTP requests (GET, POST, PUT and DELETE), through which the resources can be searched, queried, modified, and (logically) deleted. For this purpose, all queries are divided into three parts by means of RESTful operations, separate from the command itself. The first defines the base address (URL). The second identifies the primary target, which is the specific resource type such as *Patient*. Finally, the request is supplemented with parameters to filter the results:

```
GET [base]/[resourcetype]?[parameter]
```

2.6 "Extensions"

Because the central design principle of FHIR is to include only the core attributes for resources in the base standard that are allowed by the 80:20 rule, extensions become an important part of the standard to transfer data that was not considered in the development of the base model. Fig. 7 shows an example of the extension for the specific time for date of birth in the *Patient Resource*.

```xml
<birthDate value="2017-05-15">
  <extension
      url="http://hl7.org/fhir/StructureDefinition/patient-birthTime">
    <valueDateTime value="2017-05-15T17:11:00+01:00"/>
  </extension>
</birthDate>
```

Fig. 7: Example Extension for an Attribute of an Instance [14]

However, this example also explains the problems associated with extensions: For example, the extension could contain a different date. Or there may be two different extensions for the same purpose. Which extension takes precedence in such a case or how can it be converted between multiple extensions if necessary? These questions must be answered precisely by means of an implementation guide or prevented by the definition of specific constraints.

2.7 Maturity Model

FHIR introduced a new mechanism to assess the maturity or stability of a resource. This is necessary because the FHIR framework is constantly evolving and not all artifacts have an equal development history. To ensure backward compatibility, the FHIR community must take great care in implementing changes to resources that are mature. Therefore, only resources with a certain maturity level are declared normative. On the other hand, this mechanism ensures that not every request immediately leads to a new attribute, after all, most additions can first be investigated as an extension and their urgency or necessity can be identified.

2.8 Check List

With the introduction of FHIR, a number of questions (and answers) and specific requirements have arisen that must be addressed. For implementers, these details are recorded in a checklist, so that the medical security of information processing is guaranteed, and no serious errors happen due to omissions in the implementation. [4].

2.9 Using Vocabularies

Like all other standards for data exchange, FHIR must support the transmission of coded information. Many attributes in each resource require a definition for allowed codes. This is achieved by one of four data types: *Code*, *Coding*, *CodeableConcept*, and *Quantity*. They differ in principle in their flexibility as well as in the binding strength. Vocabulary is defined and provided as value sets.

The resource definitions themselves, or later within implementation guides, define the binding for each data item to an appropriate vocabulary.

The use of human-readable URLs instead of cryptic OIDs is an improvement within FHIR. These URLs make use of so-called canonical URLs for their identification, with the help of which developers can download specific content directly.

3 Conformance by Profiling

FHIR provides only the framework that everyone must adhere to in order to claim conformance to FHIR. FHIR itself contains virtually no basic requirements, so almost all attributes are optional. It provides a framework that can handle many diverse use cases. Once users have a use case definition they assemble the resources necessary to meet the use case. The base model provides this basic framework that can be adapted as needed. In other standards, some attributes are declared “required” or “mandatory”. Both have advantages and disadvantages. In concrete terms, FHIR thus becomes concretely applicable only through profiles and implementation guides, in which these additional requirements are defined more precisely, like attributes that must be supported. Therefore, whether an application has to provide the patient's name depends on the corresponding profiles.

3.1 Conformance Methodology

Making use of the FHIR framework will lead to creating a relatively large set of different profiles, which are also related to each other or can be derived from each other [2]. In the worst-case scenario, each FHIR server must support a variety of different use cases along with appropriate extensions that are managed as profiles or profile components. The HL7 Conformance Work Group was originally able to ensure that FHIR includes a compliance mechanism and that the provision of profiles and compliance resources becomes mandatory for servers. That differentiates FHIR from any other HL7 specification that does not require such a behavior. FHIR-compliant servers must provide these details online to make it easier and more independent for users to solve problems with the servers based on their real capabilities.

These profiles are technically also FHIR resources, which greatly simplifies their use.

Compliance resources can control how the server implements requests, as well as the handling of specific resources and extensions. Conversely, a FHIR server can validate the transmitted data on this basis to see if it meets the requirements.

As a central collection point for derived (e.g., country-related) resources and profiles, Simplifier.net (https://simplifier.net) was established alongside the FHIR Registry (https://registry.fhir.org, https://hl7.org/implement/standards/fhir/registry/index.html). For example, the German FHIR basic profiles (https://simplifier.net/organization/hl7deutschlandev/~projects), which are currently under further development and were used in a guide of the KBV [5] for specifying the implementation requirements according to SGB V section 1, sentence 1a, are stored on Simplifier.net.

3.2 Implementation Guides

FHIR provides the framework in which all resources are fundamentally defined. Profiles are used to mandate specific attribution of instances and rules for them. An implementation guide, which itself is a resource, is the bracket that holds everything together and also provides other implementation-relevant details.

3.3 "mustSupport"

In addition to the explicit use of compliance resources, specific implementations may differ in their support for certain FHIR functionality, such as the inclusion of resources versus references and whether and which extensions (and modifier extensions) are used.

For example, for a patient, the name, (administrative) gender and date of birth are defined, but not required for implementation. Less relevant attributes such as religion are not specified at all. Whether or not an application needs to support a specific resource or its attributes is specified by the must-support flag, which is first set at the profile level.

What is then specifically expected of an application in the event of support is described in more detail in the Implementation Guide.

To control this, a provider must explain exactly what its interface supports and expects. This declaration is made by means of conformance statements. FHIR has defined a number of resources for this purpose:

- CapabilityStatement
- ElementDefinition
- StructureDefinition
- OperationDefinition
- SearchParameters
- GraphDefinition
- ValueSet
- NamingSystem

3.4 Profiling by "Slicing"

The mechanism to get more precision (constraints) into repeatable elements in FHIR is called "slicing". Slicing results in specialized profiles as shown in Fig. 8 and can be compared with the unfolding of recursive relationships in HL7 version 3.

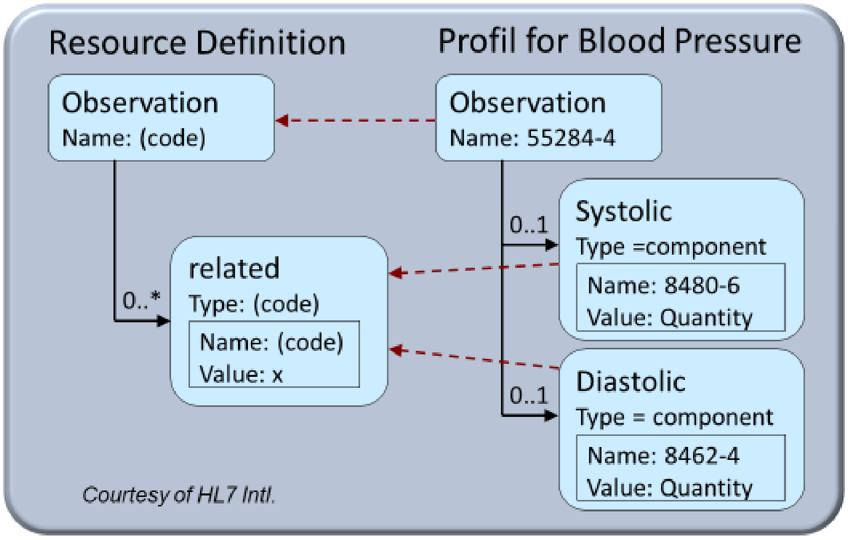

Fig. 8: "Slicing" [15]

In our example, the generic component that allows for multiple repetitions is set to exactly two repetitions with a specific type and other attribute-level constraints. FHIR's slicing capabilities, coupled with the specification of extensions, then lead to additional profiles that are necessary to resolve a specific requirement.

4 Technical Background

In addition to resources for administrative and medical content, such as *Patient*, *Encounter*, *Condition* and *Observation*, FHIR also includes resources for technical things such as code systems, value sets, concept maps, and capability statements.

The *Structure Definition Resource* is used to capture the content of the specification itself, i.e. the resources, data types, infrastructure elements, and extensions for FHIR. The *Element Definition Resource* is the means to provide these elements and details by FHIR itself.

The *Definition Resource* operation can be used to formally describe what additional functions can be provided by a particular resource and then accessed/used on the FHIR server. This includes the necessary parameters as well as the return values.

Another resource, the *Search Parameter Resource*, specifies the metadata that describes the search function and the usable parameters when searching or filtering for specific resource instances. This feature is another powerful mechanism to enhance FHIR implementations.

5 Community + Tooling

Another "lesson learned" is the support of the community through appropriate collaboration tools and platforms such as Zulip (https://chat.fhir.org), where more than 1000 interested parties present their questions, solve problems and discuss the further development of FHIR. Not to mention the many telephone conferences of working groups in order to actively get involved directly and to avoid an extensive back and forth. This also includes publicly available libraries for various development environments in Java [6] or .NET [7], profiling tools like Forge, website generation with IG publisher, as well as reference implementations and public servers [8] against which applications can be directly tested. Before the Working Group Meetings take place, peer-to-peer connect-a-thon tests are conducted, while Developer Days – DevDays for short – are used in the US and the Netherlands for training and extensive testing.

6 Reference Architecture

FHIR takes advantage of the long history of various HL7 product lines to consolidate the best properties. Thus, FHIR technically ranks between HL7 v2.x, V3 and CDA (see Fig. 1). FHIR does not contain a formal information model nor reference architecture (Fig. 9) [9]. The basics of this are only indirectly incorporated into the definition of resources, thus enabling a comprehensive exchange of data. Due to the lack of a reference architecture, a predetermined aggregation of components is not possible, and much is left to the individual requirement by facilitating existing references, even if the *Graph Definition Resource* is attempted to counteract.

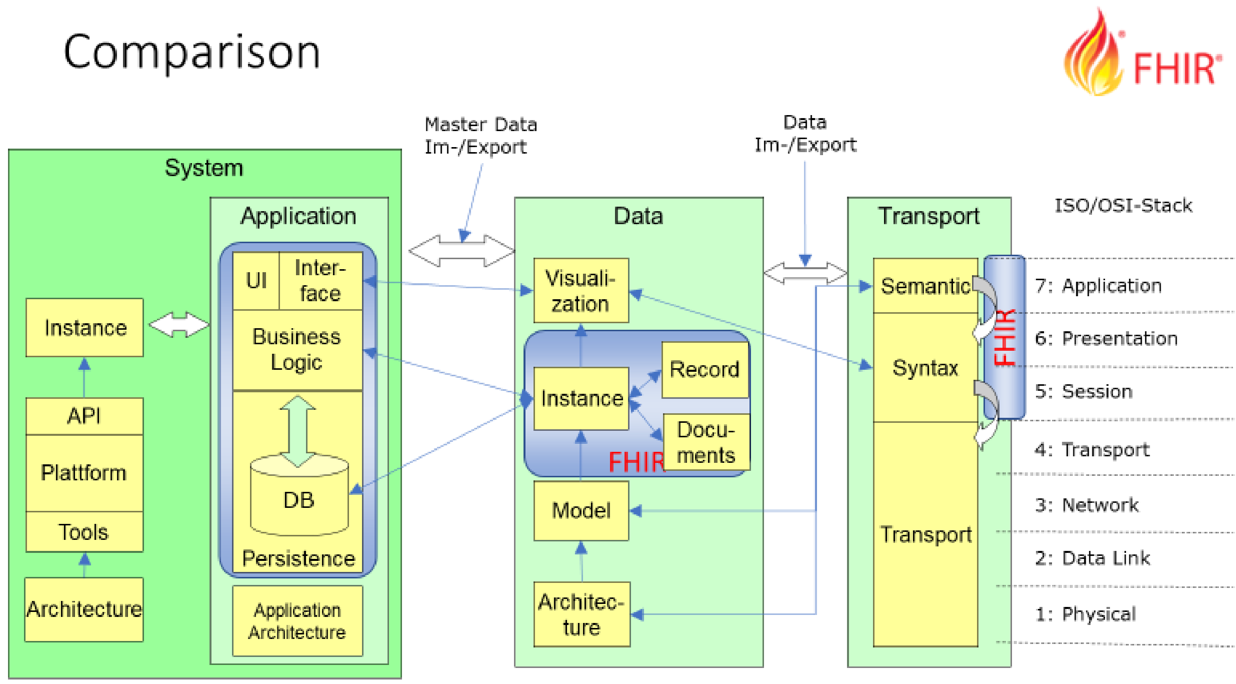

Fig. 9: Fast Healthcare Interoperability Resources (FHIR)

7 Summary

FHIR is the consistent continuation and further development of standardization activities integrating and involving many actors and their knowledge. The "lessons learned" of the past 30 years have been taken up and brought together in a new standard framework, making interoperability of many applications easier, but not yet automatic.

FHIR is thus rightly developing into "the" standard for data exchange in health and will continue to grow in importance worldwide due to the comprehensive (inter)national support of individuals and tools and will eventually have outnumbered HL7 v2.x. In Germany, more and more institutions and organizations are working on this as well and are developing guides [5] and profiles (https://simplifier.net/organization/hl7deutschlandev/~projects), some of them with official mandate. Worldwide, more than 100,000 individual resources, 17,000 profiles, extensions and guides are already stored on Simplifier.net, most of which is certainly the result of experiments, which nevertheless – or precisely because of this – confirms the success story.

However, it should not be forgotten that these individual specifications are ultimately all to be implemented. As Grahame Grieve, one of FHIR's fathers, said at the Working Group Meeting back in September 2012, "the complexity must reside somewhere". In this respect, it is not important to develop as many profiles and profile components as quickly as possible, but as few as feasible, so that the complexity of combinatorics remains manageable. This requires good documentation – and more importantly – coordination and cooperation between all actors.

A more comprehensive presentation and explanation with further details will soon be available via www.oemig.de/fhir.

8 Acknowledgement

I wish to express my sincere appreciation to Sheryl Taylor and Rob Snelick for their kind support to improve my English.

9 Literature and Links

[1] HL7 Fast Healthcare Interoperability Resources (FHIR), [Letzter Zugriff 15.06.2021]. Verfügbar unter: www.hl7.org/fhir

[2] Oemig F, Snelick R. Healthcare Interoperability Standards Compliance Handbook - Conformance and Testing of Healthcare Data Exchange Standards. Springer, 2016. ISBN 978-3-39-44837-4

[3] FHIR Exchange Module. [Letzter Zugriff 15.06.2021]. Verfügbar unter: http://www.hl7.org/implement/standards/fhir/exchange-module.html

[4] FHIR Clinical Safety. [Letzter Zugriff 15.06.20121]. Verfügbar unter: https://www.hl7.org/fhir/safety.html

[5] Kassenärztliche Bundesvereinigung (KBV). MIO Mediziische Informationsobjekte. [Letzter Zugriff 15.06.2021]. Verfügbar unter: , https://mio.kbv.de

[6] HAPI FHIR. [Letzter Zugriff 15.06.2021]. Verfügbar unter: https://hapifhir.io

[7] Firely .NET SDK. [Letzter Zugriff 15.06.2021]. Verfügbar unter : https://github.com/FirelyTeam/firely-net-sdk

[8] Test Servers. [Letzter Zugriff 15.06.2021]. Verfügbar unter: https://confluence.hl7.org/display/FHIR/Test+Servers

[9] Oemig F, Helmer A, Birkle M, Blobel B. Lösungsansätze für Interoperabilität in der Medizin. e-Health-com. Ausgabe 5/2018. p.42ff, https://e-health-com.de/fileadmin/user_upload/dateien/Downloads/EHC_5_2018_Serie_Standards__7_Langversion.pdf

[10] FHIR Patient Resource. [Letzter Zugriff 15.06.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient.html,

[11] FHIR Patient-example.json. [Letzter Zugriff 15.06.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient-example.json.html, http://www.hl7.org/fhir/patient-example.XML.html

[12] FHIR Patient-example.xml. [Letzter Zugriff 15.06.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient-example.XML.html

[13] FHIR Observation Resource. [Letzter Zugriff 15.06.2021]. Verfügbar unter: www.hl7.org/fhir/observation.html

[14] FHIR Extension: birthTime. [Letzter Zugriff 15.06.2021]. Verfügbar unter: www.hl7.org/fhir/extension-patient-birthtime.html

[15] Profiling FHIR. [Letzter Zugriff 15.06.2021]. Verfügbar unter: https://www.hl7.org/fhir/profiling.html

[16] General Procedure Example. [Letzter Zugriff 15.06.2021]. Verfügbar unter: https://www.hl7.org/fhir/procedure-example.json.html