HL7® FHIR® – Fast Healthcare Interoperability Resources: Eine Einführung

HL7® FHIR® – Fast Healthcare Interoperability Resources: an introduction

Abstract

After more than 20 years of experience with HL7 v2.x and 15 years with Version 3.0, HL7 International (http://www.hl7.org/) 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 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.

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

Frank Oemig[1,2,3]

1 Deutsche Telekom Healthcare and Security Solutions GmbH, Bonn, Deutschland

2 HL7 Deutschland, Berlin, Deutschland

3 bvitg, Berlin, Deutschland

1 Einleitung

Die HL7 Version 3.0 hatte bis zum Jahre 2009 trotz massiver Investitionen in die Entwicklung der Grundlagen, Methodiken und Werkzeuge keinen signifikanten Durchbruch am Markt erzielt. Über diese fundamentalen und begleitenden Entwicklungsschritte ist ein konkretes Verständnis entwickelt worden, wie mit Standards umzugehen ist, wie diese vom Prozessmodell her entwickelt werden müssen, wie wichtig eine Architekturgrundlage ist, und dass Datentypen, Informationsmodelle und Vokabularien zu trennen sind. Damit ist gleichzeitig klargeworden, dass eine korrekte Einhaltung aller dieser Mechanismen zusammen eine Komplexität besitzt, die auf dieser Ebene nicht zu bewältigen ist.

Parallel dazu gab es diverse Fortschritte im Bereich der Smartphones und der dazugehörigen Apps, auch war ein Trend von SOAP zu REST, von XML zu JSON sowie diverse andere parallele Entwicklungsschritte im Bereich der Kommunikationstechnologie erkennbar.

Deshalb hatte HL7 International die Frage gestellt, wie ein Standard „heute“, also 2009, aussehen müsste, wenn man mit dem aktuellen Wissen noch einmal auf der grünen Wiese anfangen könnte. Dies führte zu der sogenannten „Fresh Look Initiative2, die dieser Frage nachgehen sollte, indem sie sich mit den relevanten Arbeitsgruppen bei HL7 International darüber verständigte und die Ergebnisse anschließend zusammenführte. Diese Taskforce kam dann 2011 mit den „Resources for Healthcare“ zurück, woraus dann 2012 die „Fast Healthcare Interoperability Resources“ – abgekürzt als „FHIR“ und ausgesprochen als „fire“ – wurden und die das Beste aus den bereits existierenden Standards enthalten. Dieses neue Framework soll Schnittstellenspezifikationen für den Einsatz mit unterschiedlichen Systemen vereinfachen und modernisieren.

Abbildung 1 veranschaulicht den Ableitungsprozess sowohl in der zeitlichen Reihenfolge als auch die Einordnung der Komplexität. Das FHIR Framework selbst ist durch die Zusammenführung verschiedener Aspekte wie Datentypen, Darstellung der technischen Details, der direkten

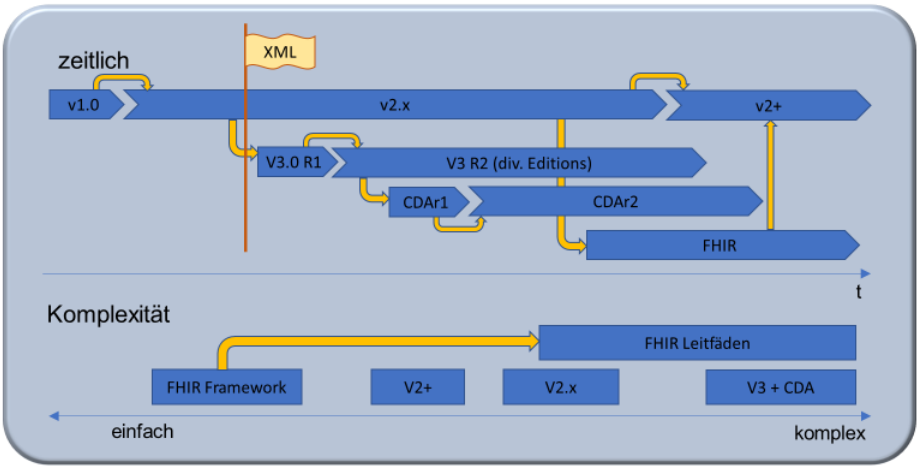

Abbildung 1: Einordnung von FHIR in die Familie der HL7 Standards

Einbindung der technischen Realisierung in Form von XML und JSON und der Angabe von vielen Beispielen (s. Abbildung 2) von der Komplexität einfacher als bspw. die anderen Standards HL7 v2.x sowie Version 3.0 mit CDA. Nach der ersten Version (Draft) als DSTU 1 (2014) wurden weitere Revisionen als Weiterentwicklung herausgegeben. Seit 2019 gibt es eine Version (R4), die erstmals normative Elemente enthält. Aktuell (1/2021) wird an R4B sowie R5 gearbeitet, ein Balloting mit Kommentierung und Abstimmung ist für Februar 2021 geplant. Für Nachfolgereleases wird erwartet, dass hier noch weitere Versionen hinzukommen.

FHIR stellt den Rahmen, innerhalb dessen man sich bewegen muss und der immer noch weiterentwickelt wird. Für eine FHIR-Konformität darf an dem Framework selbst keine Änderung vorgenommen werden. Alle zusätzlichen für eine Implementierung/Umsetzung notwendigen weiteren Details und Anforderungen werden in Implementierungsleitfäden beschrieben, die separat ausgearbeitet werden.

2 Grundlagen

Wie in der FHIR-Spezifikation [1] angegeben, werden die besten und bewährten Aspekte aus den HL7-Standards v2.x, V3 und CDA genutzt. Da wäre als wichtigster Aspekt die Komponenten-orientierung („Bausteine“) zu nennen, die in Form von „Ressourcen“ in FHIR enthalten sind. Dazu kommt das REST-Designprinzip für http-basierte Übermittlung der Daten.

Ein Grundproblem in der Standardentwicklung ist die Vollständigkeit der Spezifikation. In HL7 Version 2.9 sind derzeit knapp 2.500 Datenelemente enthalten, von denen aber nur ein ganz kleiner Teil häufig verwendet wird. Die meisten Datenelemente dienen speziellen Sonderfällen, die relativ selten umgesetzt bzw. genutzt werden. Dem wollte FHIR vorab begegnen, indem nur diejenigen Datenelemente oder Attribute festgelegt werden, für die es in 80% der eingesetzten Fälle auch einen Bedarf gibt. Damit wird der Diskussions- und Spezifikationsaufwand massiv reduziert und somit beschleunigt. Gleichzeitig sollte ein Mechanismus, der später unter der Überschrift „Extension“ erläutert wird, verhindern, dass ein Wildwuchs entsteht, wie es bspw. mit den HL7 v2.x Z-Segmenten geschehen ist. Damit wurde einerseits eine logische Öffnung erreicht, die aber technisch trotzdem geschlossen ist. Trotz dieser „Restriktion“ enthält FHIR derzeit (R4) ca. 150 Bausteine (Ressourcen) mit jeweils einer ansehnlichen Zahl an Attributen, die addiert auch schon weit im vierstelligen Bereich sind.

FHIR nutzt für die Spezifikation Mini-Modelle, die neben einer hierarchischen Baumdarstellung auch in UML (Abbildung 3) sowie in XML, JSON und weiteren Varianten (Turtle, ...) in der Spezifikation direkt nebeneinander dargestellt werden und so für ein leichteres Verständnis und damit höhere Akzeptanz sorgen.

Dadurch, dass hinter FHIR kein Architekturframework steckt, das die Einhaltung bestimmter Konventionen erzwingt, lassen sich die Attributnamen selbsterklärend benennen und die hierar-chischen Strukturen vereinfachen, was zu einer leichteren Anwendbarkeit führt und damit die Akzeptanz erheblich steigert.

Die primären Bausteine in FHIR sind Ressourcen, die in zwei verschiedenen Encodings parallel zur Verfügung gestellt werden können. Ein Encoding basiert auf XML, das andere auf der *Java Script Object Notation* (JSON), zwischen beiden kann verlustfrei konvertiert werden. Abbildung 2 demonstriert, wie dieselbe Information für eine bestimmte Patientenressource (Abbildung 3) in beiden Encodings dargestellt wird. Die Entscheidung darüber, was zu nutzen ist, wird über das Attribut *content-type* im http-Header getroffen. Hierbei sollte ein FHIR-Server beide Varianten unterstützen, um die Kommunikation mit dem Client zu vereinfachen und dabei maximale Flexibilität zu gewährleisten.

Die Daten werden auf eine inzwischen stattliche Anzahl an verschiedenen Ressourcen verteilt, die sich grob an den Klassen aus HL7 Version 3 orientieren und in Kategorien geordnet sind, um verschiedene Services zu spezifizieren.

Die Ressourcen decken ganz unterschiedliche Kategorien ab, die den integrativen Charakter von FHIR betonen:

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

Abbildung 2: Beispielinstanz für eine FHIR Patient Ressource in XML und JSON [11], [12]

- administrativ: Patient, Encounter, Location, Organization, ...
- klinisch: Observation, Concern, Allergy Intolerance, Procedure, ...
- Geräte: Devices, Metric, ...
- Terminplanung: Appointment, Slot, ...
- Finanzen: Claim, Account, Coverage, ...
- Forschung: Research Study, Research Definition, ...
- Formulare: Questionnaire, Questionnaire Response, ...
- Technisch: Structure Definition, Capability Statement, Endpoint, Test Script, ...
- Vokabular: Codesystem, Value Set, ...

2.1 Hierarchie an Elementen

In FHIR können alle Artefakte im Prinzip wie in Abbildung 4 dargestellt von einem gemeinsamen abstrakten Wurzelelement abgeleitet werden. (Abbildung 4 zeigt nur einen Ausschnitt mit wenigen Beispielen, um die

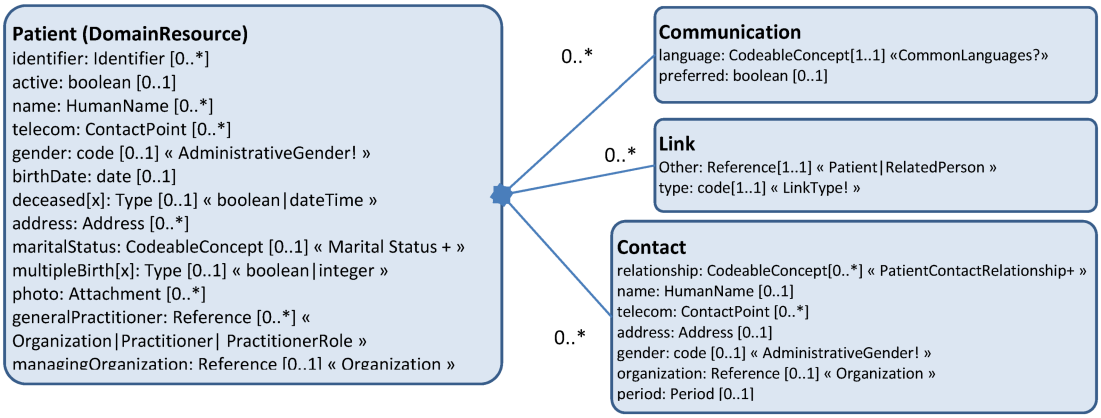

Abbildung 3: FHIR Patient Resource [10]

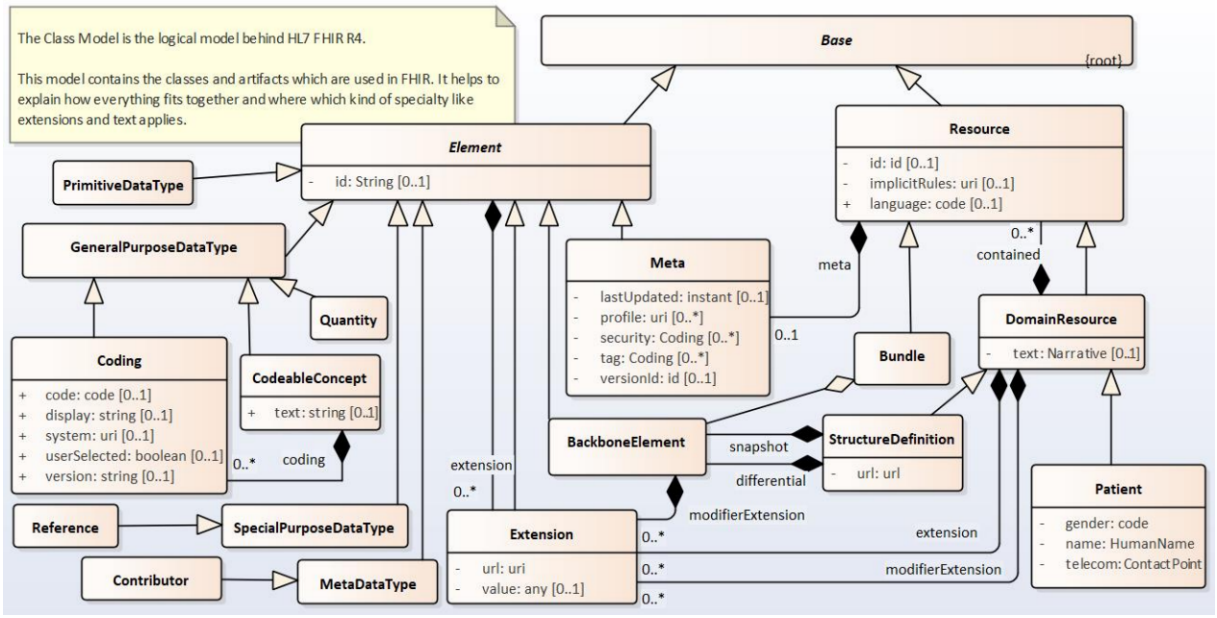

Abbildung 4: HL7 FHIR Elementhierarchie (abgeleitet aus der Spezifikation [1])

Hierarchie verdeutlichen zu können.) Daraus leiten sich dann zwei unterschiedliche Hierarchien ab, die den eigentlichen Charme von FHIR ausmachen und in jeweils der linken oder rechten Hälfte in Abbildung 4 angesiedelt sind. Das Basiselement selbst stellt die Möglichkeiten zur Verfügung, die für den Aufbau der Ressourcenhierarchie benötigt werden wie bspw. Extensions und Metadaten, so dass alle abgeleiteten Artefakte dieselben grundlegenden Eigenschaften besitzen. Gleichzeitig sorgt diese Klassenhierarchie dafür, dass diese technisch geschlossen ist und alle Details bekannt sind. Selbst die Erweiterungen („extensions“) sind auf diese Weise genaugenommen keine Erweiterungen, sondern ein vordefinierter Mechanismus für ganz gezielte Einschränkungen, die so zu Profilkomponenten werden [2].

Dazu gehören auch Datentypen. In der linken Hälfte der Abbildung 4 werden die Datentypen mit einigen Beispielspezialisierungen dargestellt. Es werden komplexe und primitive Datentypen verwendet, die sich teilweise an den Datentypen von HL7 v2.x und V3 orientieren:

- Einfach/primitiv, nur aus einem einfachen Wert bestehend
- Komplex, aber generell einsetzbar wie bspw. Namen oder Adressen
- Für Metadaten
- Spezielle Zwecke

Die für Leitfäden verwendbaren Ressourcen wie Patient und Encounter sind Spezialisierungen einer abstrakten Domain-Ressource, die wiederum auf einer abstrakten Ressourcenspezifikation basiert. Letztere stellt sicher, dass jede Ressource neben einer Identifikation für die Instanz auch Metadaten enthalten kann. Der Referenzierungsmechanismus für Ressourcen (s.u.) sowie eine optionale textliche Beschreibung der Ressource wird über die Domain-Ressourcen realisiert. Letztere ist gleichzeitig die Verankerung für Extensions sowie der sog. Modifier-Extensions. Über letztere können die Instanzen ausdrücken, dass ihre Semantik gegenüber der originalen Intention verändert wurde und deshalb einer speziellen Behandlung bedürfen.

Die Attribute einer Ressource werden entweder mit einer anderen Ressource über eine Referenz in Verbindung gebracht, oder sie sind die Grundlage für weitere Attribute

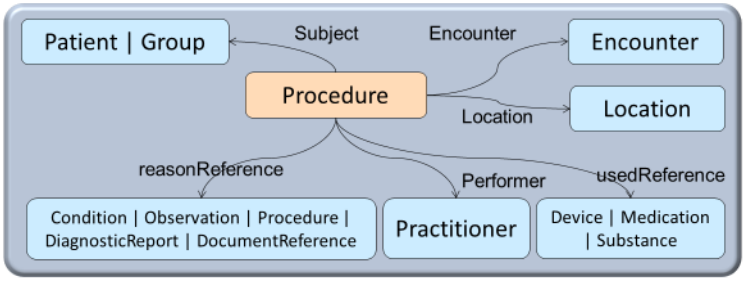

Abbildung 5: Referenzen [13]

(als Backbone-Element) oder sie werden direkt mit einem bestimmten Datentyp konkret definiert. Einige Attribute einer Ressource erlauben jedoch verschiedene Datentypen. So kann zum Beispiel das Attribut *deceased* in der Patient Resource entweder vom Datentyp Boolean oder DateTime sein, um entweder das Faktum „gestorben“ oder den genauen Todestag ausdrücken zu können (ein Zugriff auf die Information erfolgt dann in diesem Fall entweder über das Attribut *deceasedBoolean* oder *deceasedDateTime*). Eine solche Optionalität ist dann ein prädestinierter Kandidat für weitere Einschränkungen in abgeleiteten Profilen oder Implementierungsleitfäden. Es ist anzumerken, dass leere Attribute (“”) in einer Instanz nicht erlaubt sind, so dass sich die Kardinalität direkt auf das Vorkommen mit bestimmten Werten bezieht. Außerdem sind im FHIR-Framework alle Attribute optional, d.h. es gibt keine Basis-Constraints, so dass die Minimal-Kardinalität – bis auf spezielle Ressourcen – immer „0“ ist.

2.2 Grundlegende Paradigmen

FHIR unterstützt verschiedene Kommunikationsparadigmen [3]:
Die *Composition Resource* stellt mit Hilfe von Abschnitten („sections“) Dokumente bereit. Diese Sections sind zum einen geschachtelt als auch wiederholbar. Darüber hinaus können sie neben Text auch auf andere Entries über Referenzen verweisen. Wenn alle zu einem Dokument gehörenden Details zusammengestellt und übermittelt werden sollen, dann werden diese in einem Bundle (s.o.) zusammengefasst. Anders als bei CDA wird auf diese Weise anstelle einer strukturierten Hierarchie eine gleichwertige Liste von Instanzen erzeugt, die gegenseitig verweisen.
In analoger Art und Weise wird auch das Nachrichtenparadigma als *Messaging* bereitgestellt. Die dafür verantwortliche Resource ist der *MessageHeader*, der Auskunft darüber gibt, wer wem welche Nachricht warum schickt. Über eine *MessageDefinition*, auf die der *MessageHeader* verweisen kann, ist eine Präzisierung des Nachrichteninhalts möglich. Alle zu einer Nachricht gehörenden Details werden dann ebenfalls über ein Bundle zusammengefasst und übermittelt.
Das RESTful API, das für Representational State Transfer steht und mit Hilfe von zustandslosen Operationen die CRUD-Services (Create, Read, Update und Delete) realisiert, ist die Grundlage für FHIR [3]. Die Ressourcen von FHIR stellen die syntaktische Grundlage dar, mit der Daten verpackt und per http-Request übermittelt werden.
Das FHIR-Framework bietet zum einen standardisierte Basisdienste an, zum anderen stellt es über die *Operation Definition* eine Möglichkeit bereit, eigene Dienste zu definieren und anzubieten. Damit wird eine von Experten bei HL7 schon seit Jahren kritisierte Schwachstelle beseitigt, über die sich konkrete Aktionen ausführen und so bspw. das Expandieren eines Value Sets aus einer Definition oder einen Medikamentenwechselwirkungscheck realisieren lassen.
FHIR kann nicht nur als Fassade zu einer konventionellen Anwendung wie bspw. ein KIS- oder PVS-System genutzt werden, die FHIR Instanzen der Ressourcen können auch „as is“ auf sogenannten FHIR-Servern gespeichert werden. Bei letzterem bleiben dann alle Information erhalten, während bei einer Fassade nur die Details genutzt werden, die vom Server konkret unterstützt („must Support“, s.u.) werden.

2.3 Verknüpfung über Referenzen

Die FHIR Ressourcen sind die Bausteine, um komplexere Sachverhalte darzustellen. Der Standard gibt hier relativ wenig vor, wie dies genau zu geschehen hat. Hierzu nutzt FHIR eine fundamentale Eigenschaft der Web-Technologie, welche eine Instanz einer Ressource mit einer anderen Instanz durch eine Referenz in Beziehung setzen lässt.
Daher enthalten fast alle Ressourcen Attribute, die als Referenzen deklariert sind. Abbildung 5 illustriert als Beispiel die Ausprägung und Nutzung einiger Referenzen, die in der *Procedure* Ressource definiert sind.
Jede Referenz kann dabei aus einer URL und einer Textbeschreibung für eine Anzeige bestehen. Abbildung 6 zeigt ein Beispiel, wie eine solche Referenz in XML dann konkret ausgedrückt werden kann:

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

Abbildung 6: Beispielreferenz [11], [12]

Die Verwendung von Referenzen erlaubt eine sehr flexible Nutzung: So können Referenzen als absolute Referenz auf Ressourcen irgendwo im Web zeigen, oder relativ von

```
<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>
```

Abbildung 7: Beispiel für eine Extension in einem Attribut einer Instanz [14]

der Ressource, von der sie referenziert wird (innerhalb eines Servers), oder intern (in einer speziellen Zusammenstellung). Auf diese Weise kann sehr einfach ein Netz von Ressourcen aufgebaut werden, welches über mehrere Organisationen verteilt aufgespannt werden kann. So kann ein System als Provider der administrativen Daten auftreten, während ein anderes nur Befunde bereitstellt. Anstelle einer Referenz können die Ressourcen aber auch direkt eingebettet *(contained)* vorkommen, wenn sie nicht als eigenständige Instanz benötigt werden. Das reduziert den notwendigen Verwaltungsaufwand weiter.

2.4 Bundling

Der primäre Baustein in FHIR ist eine Ressource. Aber nur eine einzige Ressource zu übermitteln oder zu nutzen, reicht nur in einfachen Kontexten aus. Im einfachsten Fall kann jede Ressource alleine genutzt werden, was dann aber zu einer umfangreichen Kommunikation führt, da für konkrete Anwendungsfälle immer mehrere Ressourcen – desselben oder eines anderen Typs – betroffen sind. Deshalb müssen in der Regel mehrere Ressourcen zusammengefasst werden, um einen bestimmten Kontext herzustellen:

- Zusammenstellung mehrerer Aktionen für Ressourcen (Transaktion)
- Zusammenstellung von mehreren Abfrageergebnissen in einer Liste
- Kombination von Details, bspw. alle Werte eines Laborbefundes
- Aufbau einer Historie
- Detailergebnisse zu einem Dokument in strukturierter Form
- etc.

Solche Ressourcen werden von einer speziellen Ressource namens *Bundle* zusammengefasst. Damit können einfache Datenzusammenstellungen als simple Liste oder auch für komplexere Transaktionen realisiert werden.

2.5 http-Requests

Der primäre Datenaustausch zwischen Kommunikationspartnern erfolgt über Standard-HTTP-Requests (GET, POST, PUT und DELETE), worüber die Ressourcen gesucht, abgefragt sowie geändert und (logisch) gelöscht werden können. Dafür werden alle Abfragen mittels RESTful Operationen in drei Teile – abgesehen von dem Befehl selbst – geteilt. Der erste legt die Basisadresse („base“, URL) für einen Zugriff fest. Der zweite identifiziert das primäre Ziel, wonach gefragt wird. Dies ist dann die konkrete Ressource (*resourcetype*). Schließlich wird die Anfrage noch mit Parametern ergänzt, um die Anfrage zu präzisieren:

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

2.6 Erweiterbarkeit („Extensions“)

Da das zentrale Gestaltungsprinzip von FHIR darin besteht, nur die Kernattribute für Ressourcen in den Basisstandard mit aufzunehmen, welche durch die 80:20-Regel zugelassen werden, werden Erweiterungen zu einem wichtigen Bestandteil des Standards, um auch die Daten übertragen zu können, die bei der Entwicklung des Basismodells nicht berücksichtigt worden sind. Als Beispiel (Abbildung 7) ist die Erweiterung für die Übermittlung des konkreten Geburtszeitpunkts in der Patient Resource angeführt.
Dieses Beispiel erläutert aber auch gleichzeitig die Problematik, die mit Extensions verbunden sind: So könnte die Erweiterung ein anderes Datum enthalten. Oder es kann zwei verschiedene Extensions für denselben Zweck geben. Welche Extension bekommt in einem solchen Fall den Vorrang bzw. wie kann gegebenenfalls zwischen mehreren Extensions konvertiert werden? Diese Fragen müssen mit Hilfe eines Implementierungsleitfadens präzise beantwortet oder durch die Festlegung von konkreten Constraints verhindert werden.

2.7 Reifegrad (Maturity Model)

Mit FHIR wurde ein neuer Mechanismus eingeführt, um den Reifegrad bzw. die Stabilität einer Ressource zu bewerten. Dies ist notwendig, weil das FHIR-Framework sich kontinuierlich weiter-entwickelt und nicht alle Artefakte eine gleichlange Entwicklungsgeschichte aufweisen. Um Rückwärtskompatibilität sicherzustellen, muss die FHIR-Community große Sorgfalt in der Umsetzung von Änderungen an Ressourcen, die einen hohen Reifegrad haben, walten lassen. Daher werden nur Ressourcen mit einem bestimmten Reifegrad als normativ deklariert. Auf der anderen Seite führt dieser Mechanismus dazu, dass nicht sofort jede Anfrage zu einem neuen Attribut führt, schließlich können die meisten Ergänzungen erstmal als Extension ausprobiert und deren Dringlich- bzw. Notwendigkeit eruiert werden.

2.8 Checkliste

Mit der Einführung von FHIR traten eine Reihe von Fragen (und Antworten) sowie spezielle Anforderungen auf, die es einzuhalten gilt. Für Implementierer sind diese Details

in einer Checkliste festgehalten, damit die medizinische Sicherheit der Informationsverarbeitung gewährleistet ist und durch Auslassungen bei der Umsetzung keine gravierenden Fehler passieren [4].

2.9 Verwendungen von Vokabularien

Wie alle anderen Standards für den Datenaustausch muss auch FHIR die Übermittlung von kodierten Informationen unterstützen. Viele Attribute in den einzelnen Ressourcen benötigen eine Festlegung, welche Codes erlaubt sind. Dies geschieht jeweils über einen von vier Datentypen: *Code, Coding, CodeableConcept, Quantity*.
Diese unterscheiden sich im Prinzip in der Flexibilität sowie in der Bindungsstärke an bestimmte Vokabularien, welche als Value Sets die Code-Systeme mit den Codes festlegen und zur Verfügung stellen.
Die Bindung an geeignete Vokabularien wird für jedes Datenelement in der jeweiligen Ressourcendefinition – oder auch später im Implementierungsleitfaden – vorgenommen.
Die Verwendung von menschenlesbaren URLs anstelle der kryptischen OIDs ist eine Verbesserung durch FHIR. Diese URLs sollten als Identifikation über sogenannte canonical URLs verfügen, die auf einen konkreten Inhalt verweisen, so dass Entwickler diesen direkt herunterladen können.

3 Konformität durch Profiling

FHIR stellt nur das Rahmenwerk zur Verfügung, an das sich alle halten müssen, um zu FHIR konform zu sein. FHIR selbst enthält so gut wie keine Basisanforderungen, so dass fast alle Attribute optional sind. Auf der einen Seite ermöglicht das so einen absolut flexiblen Einsatz, auf der anderen Seite gibt es keine Werte, auf die man sich verlassen kann. In anderen Standards sind einige Attribute als required oder mandatory deklariert. Beides hat Vor- und Nachteile. Konkret anwendbar wird FHIR damit erst durch Profile und Implementierungsleitfäden, in denen diese Zusatzanforderungen wie bestimmte zu unterstützende Attribute genauer definiert werden. Ob also eine Anwendung dem Patienten einen Namen geben muss, hängt dann von den dazugehörigen Profilen ab.

3.1 Konformitätsmethodik

Durch diese Vorgehensweise entsteht ein relativ großer Satz an unterschiedlichen Profilen, die auch noch zueinander in Beziehung stehen oder voneinander abgeleitet sein können [2]. Im schlimmsten Fall muss jeder FHIR-Server eine Vielzahl von verschiedenen Ausprägungen zusammen mit entsprechenden Erweiterungen unterstützen, die als Profile oder Profil-komponenten verwaltet werden. Die HL7 Conformance Work Group konnte ursprünglich sicherstellen, dass FHIR einen Konformitätsmechanismus beinhaltet und die Bereitstellung von Profilen und Konformitätsressourcen für die Anbieter verpflichtend wird. Damit unterscheidet sich FHIR gravierend von allen anderen HL7-Spezifikationen, die das nicht fordern. FHIR-konforme Server müssen die vorab aufgeführten Erklärungen online zur Verfügung stellen, damit die Anwender Probleme im Umgang mit den Servern auf Basis der Profile leichter und unabhängiger lösen können.
Bei diesen Profilen handelt es sich technisch ebenfalls um FHIR-Ressourcen, was deren Verwendung damit stark vereinfacht.
Die Konformitätsressourcen können kontrollieren, wie der Server das Suchen sowie bestimmte Ressourcen und Erweiterungen implementiert. Ein FHIR-Server kann umgekehrt auf dieser Basis die übermittelten Daten validieren, ob diese den Anforderungen genügen.
Als zentrale Sammelstelle für abgeleitete (z.B. länderbezogene) Ressourcen und Profile scheint sich bislang Simplifier.net (https://simplifier.net/) neben der FHIR Registry (https://registry.fhir.org/, http://hl7.org/implement/standards/fhir/registry/index.html) zu etablieren. Bei ersterem sind bspw. auch die deutschen FHIR-Basisprofile (https://simplifier.net/organization/hl7deutschlandev/~projects) hinterlegt, die sich derzeit in weitere Ausarbeitung befinden und in einem Leitfaden der KBV [5] zur Umsetzung der Vorgaben nach SGB V §291d Abst. 1, Satz 1a verwendet wurden.

3.2 Implementierungsleitfäden

FHIR stellt das Rahmenwerk, in dem alle Ressourcen grundlegend definiert sind. Über Profile werden konkrete Instanzausprägungen sowie Regeln dazu vorgeschrieben. Ein Implementierungsleitfaden, welcher selbst eine Ressource darstellt, stellt dazu die Klammer dar, die alles zusammenhält und darüber hinaus noch weitere umsetzungsrelevante Details bereitstellt.

3.3 „mustSupport“

Neben der ausdrücklichen Nutzung von Konformitätsressourcen können sich konkrete Implementierungen in ihrer Unterstützung bestimmter FHIR-Funktionalität unterscheiden, wie zum Beispiel die Inklusion von Ressourcen im Vergleich zu Referenzen und ob und welche (Modifikator-)Erweiterungen (modifierExtension) verwendet werden. Für einen Patienten sind zum Beispiel der Name, das (administrative) Geschlecht und das Geburtsdatum definiert, aber nicht für eine Umsetzung vorgeschrieben. Weniger relevante Attribute wie die Religion sind gar nicht erst spezifiziert. Ob eine Anwendung eine spezielle Ressource oder deren Attribute unterstützen muss oder nicht, wird über das must-Support-Flag vorgeschrieben, das erst auf Profilebene gesetzt wird.
Was dann im Falle einer Unterstützung von einer Anwendung konkret erwartet wird, wird wiederum im Implementierungsleitfaden genauer beschrieben.
Um dies zu kontrollieren, muss ein Anbieter genau erklären, was seine Schnittstelle unterstützt und erwartet. Diese Erklärung erfolgt durch sogenannte Konformitäts-

erklärungen (conformance statements). FHIR hat hierfür eine Reihe von Ressourcen definiert:

- CapabilityStatement
- ElementDefinition
- StructureDefinition
- OperationDefinition
- Search Parameters
- GraphDefinition
- Value Set
- Naming System

3.4 Profiling durch „Slicing“

Der Mechanismus, um weitere Präzision (Constraints) in wiederholbare Elemente in FHIR zu bekommen, wird „Slicing“ genannt. Slicing resultiert wie in Abbildung 8 dargestellt in spezialisierten Profilen und kann mit dem Auseinanderfalten von rekursiven Beziehungen in HL7 Version 3 verglichen werden.

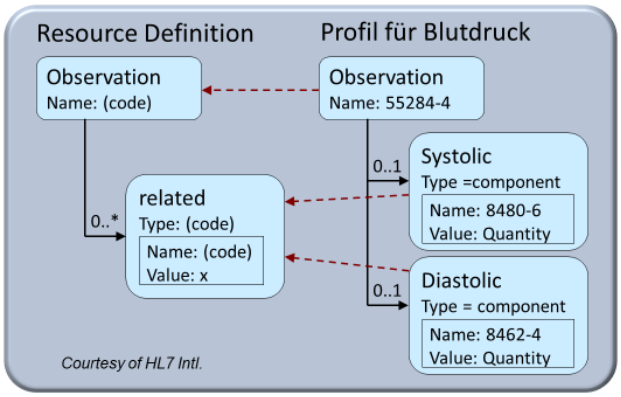

Abbildung 8: „Slicing“ [15]

In unserem Beispiel wird die generische Komponente, die mehrere Wiederholungen erlaubt, auf exakt zwei Wiederholungen mit einem spezifischen Typ und weiteren Einschränkungen auf Attributebene festgelegt. Die Slicing-Fähigkeiten von FHIR in Verbindung mit der Spezifikation von Erweiterungen führen dann zu weiteren Profilen, die notwendig sind, um eine spezielle Anforderung zu lösen.

4 Technische Hintergründe

Neben Ressourcen für administrative und medizinische Inhalte, wie beispielsweise *Patient*, *Encounter*, *Condition* und *Observation*, enthält FHIR aber auch Ressourcen für technische Dinge wie beispielsweise *Codesystems*, *Value-Sets*, *Concept Maps* und *Capability Statements*.
Die *Structure Definition Resource* wird verwendet, um den Inhalt der Spezifikation selbst festzuhalten, d.h. die Ressourcen, Datentypen, Infrastrukturelemente und Erweiterungen für FHIR. Damit können diese Elemente sozusagen direkt in FHIR ausgedrückt werden. Dazu bedient diese Resource sich der *Element Definition Resource*.
Mit der *Operation Definition Resource* kann formal beschrieben werden, welche zusätzlichen Funktionen durch eine bestimmte Ressource zur Verfügung gestellt und dann auf dem FHIR-Server aufgerufen/genutzt werden können. Das umfasst die notwendigen Parameter sowie die Rückgabewerte.
Eine weitere Ressource, die *Search Parameter Resource*, spezifiziert die Metadaten, die die Suchfunktion und die verwendbaren Parameter beschreiben, wenn nach bestimmten Ressourceninstanzen gesucht oder gefiltert werden soll. Darüber wird ein weiterer Erweiterungsmechanismus beschrieben.

5 Community + Tooling

Eine weitere „lesson learned“ ist die Unterstützung der Community durch geeignete Kollaborationstools und -plattformen wie Zulip (https://chat.fhir.org), auf der mehr als 1.000 Interessierte ihre Fragen vorstellen, Probleme lösen und die Weiterentwicklung von FHIR absprechen. Nicht zu vergessen sind dabei auch die vielen Telefonkonferenzen der Arbeitsgruppen, um sich aktiv direkt einzubringen und ein ausgiebiges Hin und Her zu vermeiden. Dazu gehören auch öffentlich verfügbare Libraries für diverse Entwicklungsumgebungen in Java [6] oder .NET [7] sowie Referenzimplementierungen und öffentliche Server [8], gegen die Anwendungen direkt getestet werden können. Vor den Working Group Meetings werden Peer-to-Peer-Tests á la Connect-a-thon durchgeführt, während die Developer Days – kurz DevDays – in den USA und den Niederlanden für Schulungen und ausgiebige Tests genutzt werden.

6 Referenzarchitektur

FHIR macht sich die lange Historie der verschiedenen HL7-Produktlinien zunutze, um die besten Eigenschaften zu konsolidieren. So rangiert FHIR technisch zwischen HL7 v2.x, V3 und CDA (vgl. Abbildung 1). FHIR enthält weder ein formales Informationsmodell noch eine Referenzarchitektur (Abbildung 9) [9]. Die Grundlagen davon fließen nur indirekt in die Definition der Ressourcen ein und ermöglichen so einen umfassenden Datenaustausch. Wegen der fehlenden Referenzarchitektur ist eine vorherbestimmte Aggregation von Komponenten nicht möglich und vieles wird durch die vorhandenen Referenzen dem jeweiligen Erfordernis überlassen, auch wenn mit der *GraphDefinition Resource* gegenzusteuern versucht wird.

7 Schlussfolgerung

FHIR ist die konsequente Fortsetzung und Weiterentwicklung der Standardisierungsaktivitäten unter Einbeziehung und Einbindung vieler Akteure und deren Wissen. Die „lessons learned“ der vergangenen 30 Jahre wurden aufgegriffen und in einem neuen Standard-Framework zusammengeführt, so dass Interoperabilität vieler Anwen-

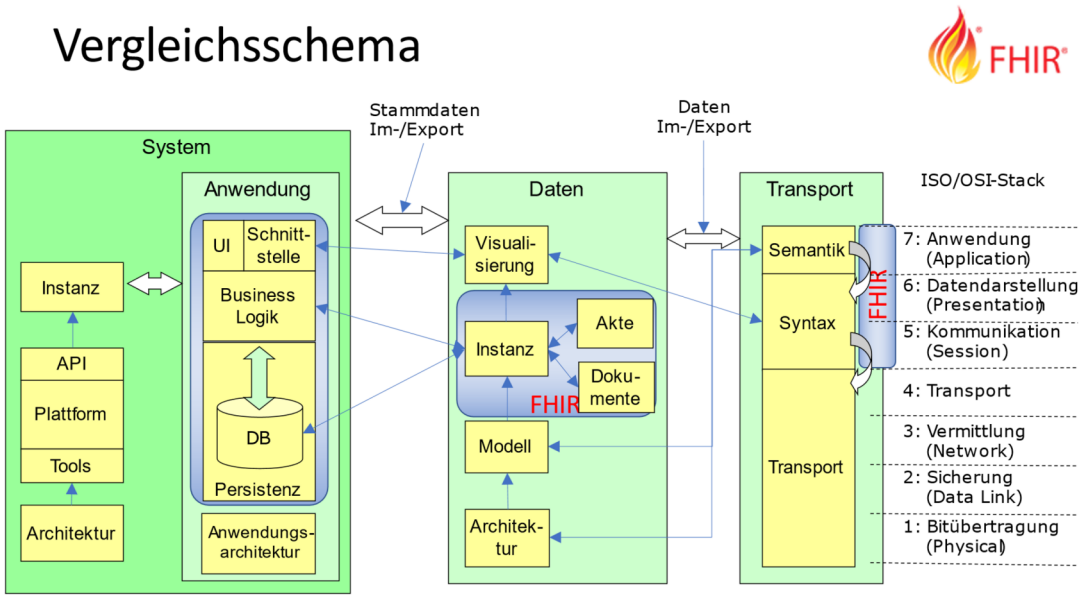

Abbildung 9: Fast Healthcare Interoperability Resources (FHIR)

dungen einfacher wird, aber damit noch nicht automatisch gegeben ist.

FHIR entwickelt sich damit zu Recht zu „dem“ Standard für den Datenaustausch im Gesund-heitswesen und wird durch die umfassende (inter)nationale Unterstützung durch Personen und Werkzeuge an Bedeutung weltweit weiter zunehmen und irgendwann HL7 v2.x den Rang abgelaufen haben. Auch in Deutschland beschäftigen sich immer mehr Institutionen und Organisationen damit und entwickeln – teilweise mit offiziellem Auftrag – Leitfäden [5] und Profile (https://simplifier.net/organization/hl7deutschlandev/~projects). Weltweit sind auf Simplifier.net schon heute mehr als 100.000 individuelle Ressourcen, 17.000 Profile, Erweiterungen und Leitfäden hinterlegt, von denen sicherlich ein Großteil das Ergebnis des Ausprobierens ist, was trotzdem – oder gerade deshalb – die Success-Story bestätigt.

Allerdings darf hierüber nicht vergessen werden, dass diese Einzelspezifikationen letztendlich auch alle umzusetzen sind. Wie sagte Grahame Grieve, einer der Väter von FHIR, schon im September 2012 auf dem Working Group Meeting, „the complexity must reside somewhere“. Insofern kommt es eher nicht darauf an, möglichst schnell möglichst viele und insbesondere eigene Profile bzw. Profilkomponenten zu entwickeln, sondern möglichst wenige, damit die Komplexität der Kombinatorik handhabbar bleibt. Das bedarf einer guten Dokumentation – und noch viel wichtiger – Abstimmung und Kooperation aller Akteure.

Eine umfassendere Darstellung und Erläuterung mit weiteren Details wird in Kürze über http://www.oemig.de/FHIR/ verfügbar sein.

Anmerkung

Interessenkonflikte

Der Autor erklärt, dass er keine Interessenkonflikte in Zusammenhang mit diesem Artikel hat.

Literatur

1. HL7 Fast Healthcare Interoperability Resources (FHIR). [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/
2. Oemig F, Snelick R. Healthcare Interoperability Standards Compliance Handbook – Conformance and Testing of Healthcare Data Exchange Standards. Heidelberg: Springer; 2016.
3. FHIR Exchange Module. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/implement/standards/fhir/exchange-module.html
4. FHIR Clinical Safety. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://www.hl7.org/fhir/safety.html
5. Kassenärztliche Bundesvereinigung (KBV). MIO Medizinische Informationsobjekte. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://mio.kbv.de
6. HAPI FHIR. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://hapifhir.io
7. Firely.NET SQK. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://github.com/FirelyTeam/firely-net-sdk
8. Test Servers. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://confluence.hl7.org/display/FHIR/Test+Servers
9. Oemig F, Helmer A, Birkle M, BlobelM. Lösungsansätze für Interoperabilität in der Medizin. e-Health-com. 2018; (5):42ff. Available from: https://e-health-com.de/fileadmin/user_upload/dateien/Downloads/EHC_5_2018_Serie_Standards__7_Langversion.pdf
10. FHIR Patient Resource. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient.html
11. FHIR Patient-example.json. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient-example.json.html

12. FHIR Patient-example.xml. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/patient-example.XML.html

13. FHIR Observation Resource. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/observation.html

14. FHIR Extension: birthTime. [Letzter Zugriff 20.01.2021]. Verfügbar unter: http://www.hl7.org/fhir/extension-patient-birthtime.html

15. Profiling FHIR. [Letzter Zugriff 20.01.2021]. Verfügbar unter: https://www.hl7.org/fhir/profiling.html

Korrespondenzadresse:
Dr. Frank Oemig
Deutsche Telekom Healthcare and Security Solutions GmbH, Friedrich-Ebert-Allee 140, 53113 Bonn, Deutschland
frank.oemig@t-systems.com

Bitte zitieren als
Oemig F. HL7® FHIR® – Fast Healthcare Interoperability Resources: Eine Einführung. GMS Med Inform Biom Epidemiol. 2021;17(2):Doc09. DOI: 10.3205/mibe000223, URN: urn:nbn:de:0183-mibe0002234

Artikel online frei zugänglich unter
https://www.egms.de/en/journals/mibe/2021-17/mibe000223.shtml

Veröffentlicht: *26.04.2021*