

## Catching FHIR

Kevin Mayfield ([www.mayfield-is.co.uk](http://www.mayfield-is.co.uk))

---

*@timbenson – The best new health interoperability standard for 20 years*

Fast Health Interoperable Resource (FHIR pronounced fire) is a new standards framework from HL7. It builds on the strength of HL7v2 using the latest web standards without the complexity of HL7v3. It is built around modular components called resources; these clinical resources should be instantly familiar to those of us who have worked with EPR or GP systems. For a list of the current FHIR resources see <http://www.hl7.org/implement/standards/fhir/resourcelist.html>

Resources can be represented either by XML or JSON, interactions via RESTful methods are most common but others are possible, FHIR is not a connection standard. Unlike previous HL7 standards which required membership to access, the FHIR specifications are open and accessible to all. FHIR is currently in draft standard for trial use (DSTU) with clinical and workflow resources stable and patient administration resources being developed.

FHIR is has implemented resources/data elements, if 80% of systems currently implement it. Rather than the 20% of data items that tend to be features or requirements from large vendors/centralised national programmes (i.e. edge cases, which could be left to HL7v3?). This is called the FHIR 80/20 rule.

### Patient Resource

To view an example Patient resource using REST we can simply navigate to a FHIR server using this URL:

<http://fhir.healthintersections.com.au/open/Patient/1>

The returned patient resource rendered as an html webpage as shown below.



The screenshot shows a web browser window with the address bar displaying `fhir.healthintersections.com.au/open/Patient/1`. The page content includes a navigation bar with links for `FHIR © HL7.org`, `Server Home`, `Health Intersections FHIR Server`, `FHIR Version 0.80-2345`, and `User: 82.31.40.119`. The main heading is `Resource 1`. Below the heading, there is a section for `Tags: (no tags) +` and a section for `XML or JSON representation`. The patient details are listed as follows:

Name	Peter James <b>Chalmers</b> ("Jim")
Address	534 Erewhon, Pleasantville, Vic, 3999
Contacts	Home: unknown. Work: (03) 5555 6473
Id	MRN: 12345 (Acme Healthcare)

Alternatively, we could create a simple method using Cache ObjectScript class. Then via Cache terminal we can return the data as XML (`d ##class(CAMTA.FHIR).GetPatient()`) or JSON (`d ##class(CAMTA.FHIR).GetPatient("json")`). [The class should work on most Cache versions]

```
Class CAMTA.FHIR Extends %RegisteredObject
{
ClassMethod GetPatient(format As %String)
{
    //d ##class(CAMTA.FHIR).GetPatient("json")
    set tHttpRequest=##class(%Net.HttpRequest).%New()
    Set tHttpRequest.Server="fhir.healthintersections.com.au"
    set tHttpRequest.Port=80
    Set tHttpRequest.Timeout=5

    if $g(format)="""
    {
        Do tHttpRequest.Get("open/Patient/1")
    }
    else
    {
        Do tHttpRequest.Get("open/Patient/1?_format=" _format)
    }

    w !,tHttpRequest.HttpResponse.StatusCode

    if tHttpRequest.HttpResponse.StatusCode=200
    {
        while tHttpRequest.HttpResponse.Data.AtEnd=0
        {
            w !,tHttpRequest.HttpResponse.Data.Read()
        }
    }
}
}
```

The parameters we used in the previous example returned a single resource but within FHIR we can alter the parameters to search for Patient resources, in the following example we search for Patient resources for ID 12345 assigned by authority represented by the OID.

[http://fhir.healthintersections.com.au/open/Patient?\\_search?identifier=urn:oid:1.2.36.146.595.217.0.1|12345](http://fhir.healthintersections.com.au/open/Patient?_search?identifier=urn:oid:1.2.36.146.595.217.0.1|12345)

This time FHIR returns an atom feed containing a list of Patient resources.

Search results for resource x

fhir.healthintersections.com.au/open/Patient?\_search?identifier=urn:oid:1

FHIR © HL7.org | Server Home | Health Intersections FHIR Server | FHIR Version 0.80-2345 | User: 82.31.40.119

## Search results for resource type Patient

Atom (XML) or JSON representation

Search Links: [First](#) [Previous](#) [Next](#) [Last](#) (438 found). Search: [http://fhir.healthintersections.com.au/open/Patient/\\_search?\\_format=text/xhtml&search-id=4a2886a1-e1be-41dc-9885-1827da03e1&search-sort=\\_id](http://fhir.healthintersections.com.au/open/Patient/_search?_format=text/xhtml&search-id=4a2886a1-e1be-41dc-9885-1827da03e1&search-sort=_id)

SQL: `Insert into SearchEntries Select 3393, ResourceKey, Id from Ids where Ids.MasterResourceKey is null and Ids.ResourceTypeKey = 37 order by ResourceKey DESC`

**Patient "05680820-78d7-422d-8585-b46153baeef7" Version "1"**

Tags: (no tags) +

[this resource](#), [XML](#) or [JSON](#) representation, or [the full version history](#). Updated: 2014-03-20T09:12:11Z; Author: 93.57.39.68

```
<?xml version="1.0" encoding="UTF-8"?>
<Patient xmlns="http://hl7.org/fhir"/>
```

**Patient "08862666-f1d2-4fa2-b1f6-a3a6ff27a7bc" Version "1"**

Tags: (no tags) +

[this resource](#), [XML](#) or [JSON](#) representation, or [the full version history](#). Updated: 2014-03-20T09:31:36Z; Author: 93.57.39.68

## Creating FHIR Resources in Cache

Creating FHIR resources in ObjectScript is also relatively simple. The schema can be obtained from

<http://www.hl7.org/documentcenter/public/standards/FHIR/fhir-all-xsd.zip>

Then imported via Cache studio in the normal manner. [In the following example, <http://hl7.org/fhir> has been mapped to **hl7.fhir** package and <http://www.w3.org/1999/xhtml> has been mapped to **xhtml** package.]

The sample method below has been built to communicate important events between health and social services in Scotland. Object identifiers (OID's) have been used to identify coding schemes between agencies; this is not mandated in FHIR.

```

ClassMethod CreateObservation()
{
    //d ##class(CAMTA.FHIR).CreateObservation()
    s tObservation=##class(hl7.fhir.Observation).%New()

    // Using the OID for NHS CHI Number with a dummy CHI reference
    s tObservation.subject = ##class(hl7.fhir.ResourceReference).%New()
    s tObservation.subject.reference = ##class(hl7.fhir.string).%New()
    s tObservation.subject.reference.value =
"urn:oid:2.16.840.1.113883.2.1.3.2.4.16.53|12345ABCDE"

    // Plain description
    s tObservation.subject.display = ##class(hl7.fhir.string).%New()
    s tObservation.subject.display.value = "Ms Janet Smith"

    s tObservation.text = ##class(hl7.fhir.Narrative).%New()
    s tObservation.text.div = ##class(xhtml.div).%New()
    s tObservation.text.div.content = "Unborn child subject to child protection plan"

    s tObservation.name = ##class(hl7.fhir.CodeableConcept).%New()

    // Concept ID for the UK specific code for 'Unborn child subject to child protection plan'
    s tCode = ##class(hl7.fhir.Coding).%New()
    s tCode.code = ##class(hl7.fhir.code).%New()
    s tCode.code.value = "818901000000100"
    s tCode.display = ##class(hl7.fhir.code).%New()
    s tCode.display.value = "Unborn child subject to child protection plan"

    //OID for the UK edition of SNOMED, we could have used http://snomed.info/sct for non UK
codes
    s tCode.system = ##class(hl7.fhir.uri).%New()
    s tCode.system.value = "urn:oid:2.16.840.1.113883.2.1.3.2.4.15"
    d tObservation.name.coding.Insert(tCode)

    // ok = The result has no reliability concerns.
    s tObservation.reliability = ##class(hl7.fhir.ObservationReliability).%New()
    s tObservation.reliability.value = "ok"

    // Adding the OID for City of Edinburgh Council
    s tOrganisationReference = ##class(hl7.fhir.ResourceReference).%New()
    s tOrganisationReference.reference = ##class(hl7.fhir.string).%New()
    s tOrganisationReference.reference.value =
"Patient?identity=urn:oid:2.16.840.1.113883.2.1.3.26"
    s tOrganisationReference.display = ##class(hl7.fhir.string).%New()
    s tOrganisationReference.display.value = "City of Edinburgh Council"
    d tObservation.performer.Insert(tOrganisationReference)

    s tObservation.issued = ##class(hl7.fhir.instant).%New()
    s tObservation.issued.value = "2014-04-16 09:31"

    s ns=##class(%XML.Namespaces).%New()

```

```

d ns.AddNamespace("http://hl7.org/fhir")
d tObservation.XMLExportToString(.tStr,,,ns)
w !,tStr
}

```

Executing the code via `d ##class(CAMTA.FHIR).CreateObservation()` returns the following XML. Unlike HL7v3 and HL7v2 it is quite easy to understand the returned resource, it has the self-descriptive properties of XML and sticks to the information that needs to be exchanged without the confusion of a complex information model.

```

<Observation xmlns="http://hl7.org/fhir">
  <text><div xmlns="http://www.w3.org/1999/xhtml">Unborn child subject
to child protection plan</div>
  </text>
  <name>
    <coding>
      <system
value="urn:oid:2.16.840.1.113883.2.1.3.2.4.15"></system>
      <code value="818901000000100"></code>
      <display value="Unborn child subject to child
protection plan"></display>
    </coding>
  </name>
  <issued value="2014-04-16T09:31Z"></issued>
  <reliability value="ok"></reliability>
  <subject>
    <reference value="
Patient?identity=urn:oid:2.16.840.1.113883.2.1.3.2.4.16.53|12345ABCD
E"></reference>
    <display value="Ms Janet Smith"></display>
  </subject>
  <performer>
    <reference
value="urn:oid:2.16.840.1.113883.2.1.3.26"></reference>
    <display value="City of Edinburgh Council"></display>
  </performer>
</Observation>

```

## Why did we choose FHIR?

FHIR is built on the proven knowledge and experience of HL7v2 standard and as can be seen from the earlier examples it is easier to understand than HL7v2 and HL7v3

Although HL7 FHIR and NHS Scotland data standards overlap, FHIR resources already have more clinical coverage and are more comprehensively documented. We have not gone for a rip and replace approach; if a NHS Scotland (or HL7v2) standard already existed and was in use, then we would continue to use that otherwise we would use the FHIR standard. (This also prevents us from having to document and manage extensions to NHS Scotland data standards)

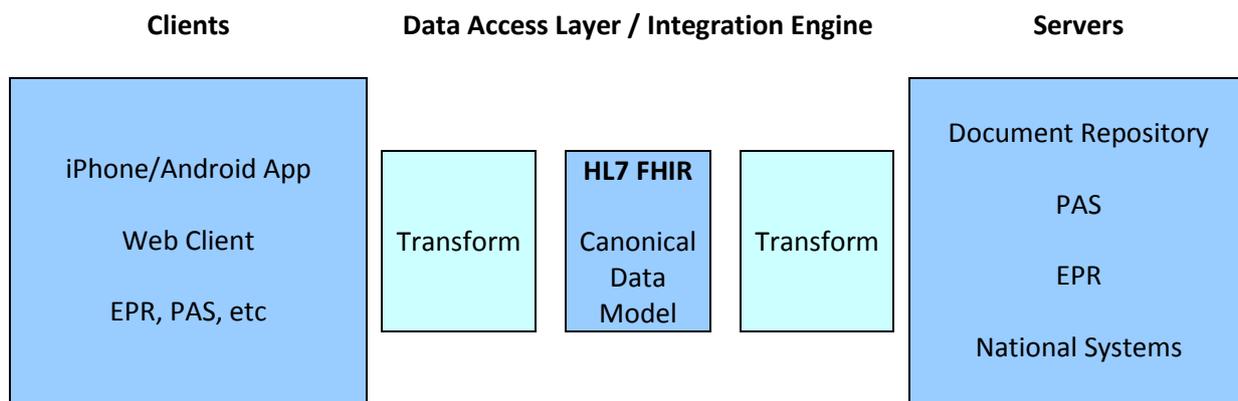
Conversion to other standards is easy from FHIR, the data model has been engineered to be compatible with other standards and patterns such as HL7v3, HL7v2, IHE PIX/PDQ/XDS/PCC, etc. This compatibility has allowed us standardise our document handling to follow IHE XDS patterns via the FHIR Binary, DocumentReference and DocumentManifest resources. This helps future proof our document handling, especially around the metadata, should we decide to move to an IHE XDS Document Repository in the future.

Using FHIR+REST would reduce the complexity and the number of interfaces. REST allowed us to reduce the number of differing complex SOAP based web services and loosened the coupling between systems. SOAP required synchronisation of WSDL files between agencies which took coordination, this would frequently occur when the web services were extended, either for new interfaces or modifications to existing ones. FHIR on the other hand has a rich data model, which more often than not would already have the 'new' element or we could bring into service an unused resource on the existing interfaces without a high maintenance cost (FHIR will also be backwardly compatible which reduces requirements to keep schemas synchronised). In addition we would be able to remove ODBC or ECP connections and communicate with Cache data stores directly using REST interfaces built on CSP pages.

## How has it worked in practice?

By far the most important change has been the rate of development. FHIR+REST fits with the techniques and skills taught in universities and/or used in other sectors. Developers have quickly learned and understood the FHIR standards at a higher level of comprehension compared to the existing standards. This has led to an increased rate of development and a higher level of integration between systems/providers.

FHIR used as a canonical data model (Enterprise Integration Patterns (EIP) – Hohpe & Woolf), has allowed interfaces to be developed in a modular fashion using multiple developers rather than the one developer which tended to be the case with Point To Point (Message Translators – EIP).



**Fig 1. Canonical Data Model**

As most systems are also producing or consuming FHIR, they can also be reused for testing new interfaces as substitute endpoints.

### *Patient Index Updates and reuse*

As part of our portal project we are taking patient resource feeds from several social services and NHS systems. Each has its own API and XML standards. To avoid this myriad of standards each system is converted to FHIR, these can be developed in isolation. The general pattern being:

- Convert to FHIR Patient resource
- Send the resource to the Health Information Exchange (HIE) using REST
- HIE transforms these and sends the messages to the portals API

Another part of the project developed a FHIR Patient validation interface (Patient/\_validate). Due to concerns over the reliability of NHS CHI numbers from some sources, this interface was incorporated into the patient index updates working flow. Resulting in the pattern below:

- Convert to FHIR Patient resource
- Send the resource to the Health Information Exchange (HIE) using REST
- *Validate the resource using the Patient/\_validate interface*
- HIE transforms these and sends the messages to the portals API

With all interfaces using the same model, this was a quick simple reuse of the existing Patient resource, passing it to the validation service and then acting on the returned response code.

Using FHIR also minimised the impact of changing the supplier of the portal. The new supplier accepted IHE PIX style updates, which only required the single FHIR Patient resource feed to be altered and was a straight forward transformation. Normal message transformation interfaces would have all the transforms to be converted, the FHIR canonical data model only required amending one.

### **Conclusion**

FHIR is simple, straightforward and so powerful. Hopefully its adoption will be swift and quickly improve patient care coordination between agencies and systems.