# FHIRed Up

If you were one of the many HL7 members who participated in the May 2012 Vancouver working group meeting, your likely encountered many references (and not a few bad puns) to something called “FHIR”. It received mention in presentations from the CEO, the Board chair, and the chair of the TSC. The FHIR tutorials on Sunday had their room size increased twice and a few of the work group meetings where FHIR was on the agenda were standing room only. So what’s causing all this excitement and interest?

FHIR (Fast Healthcare Interoperability Resources) is a new methodology for HL7. It builds on the semantic foundations of HL7 v3, while leveraging lessons learned from v2, v3 and CDA implementation experience to radically reduce the effort required to build interoperable systems and at the same time to simplify and speed the standards development process.

Before we start:

Some HL7 stakeholders have voiced a concern that the introduction of FHIR will mean the end of HL7’s commitment to other HL7 standards such as v2, v3 or CDA. There is no need to worry. HL7 will continue to support and develop v3 and other standards for so long as implementers require, just as they have continued to support HL7 v2 with the introduction of v3. The proportion of resources spent on FHIR and other initiatives will be driven by the interests and needs of the implementation community.

## Premises and design

FHIR is based on a few simple premises:

1. Interface specifications should be designed first and foremost for implementers. This includes how they’re documented and how they approach technical solutions
2. HL7 should follow current cross-industry best practices about how to do interoperability – including support for a RESTful approach
3. Specifications should standardize only those data elements and processes that are common to the bulk of implementations (“the 80%”)
4. Extensibility is an essential part of robust interoperability, but needs to be properly managed.

The original design of FHIR was based on a popular set of interoperability specifications in the customer relationship management space called 37 Signals. FHIR takes the core ideas that make a RESTful approach easy to implement, as used by many large cloud applications, and applies them to the healthcare domain. The essence of FHIR is as follows:

1. All data structures are broken up into carefully managed chunks called Resources
   1. Resources themselves are designed as XML structures with names and element nesting reflecting the way domain experts and implementers think about the content. The work group designs exactly what the implementer will see with no abstract models in the way.
2. Only those data elements expected to be used by 80% of software systems implementing the resource are candidates for inclusion as “core” elements in the resource (“the 80%”).
3. Data elements include formal mappings to the RIM and ISO 21090 datatype specifications to ensure semantic rigor, but these mappings are presented in a manner that implementers can ignore them if they wish.
   1. Mappings to HL7 v2 and other key industry specifications will also be provided.
4. All resources have the ability to carry a textual representation of the resource in place of or in addition to the structured data representation.
5. All resources have an extensibility section where additional data elements can be conveyed that don’t meet the requirements to be considered “core”.
   1. Extensions are managed as terminologies with a requirement that the extension definition be electronically accessible in a standardized form from within the environment using the extension.
   2. Extensions declare whether they must be understood or can be safely ignored by implementers.
6. Datatypes and value set definitions are also designed reflecting the “80%” approach to implementation.
7. Resources, once normative, will remain wire format backwards compatible for all future releases of FHIR.

## Resources

The core structure of FHIR is “resource”

<put stuff here>

## FHIR Interoperability Paradigms

FHIR supports a number of different interoperability paradigms: REST, messaging, documents and services.

At its core, FHIR is based on a REST approach. Every resource can be exposed as a REST interface from which resource instances can be created, retrieved, updated and “deleted”. This is a powerful, web-based approach to interoperability that has particular interest for the mobile app community. (Many “Web 2.0” systems are based on REST technologies.)

For some use cases, manipulating individual resources is not sufficient. Groups of resources need to be managed together in a transactional manner. In FHIR, this can be done with documents (where the focus is attestation and human-readability) or messaging (where the focus is workflow in a manner similar to HL7 v2). However FHIR offers the ability to combine both aspects together and package resources together with metadata reflecting both a document and messaging perspective.

Finally some use cases require more sophisticated choreographies than can be provided in a simple messaging paradigm. In these situations, service specifications can be designed that use resources

## What does this mean for implementers?

Nothing – yet. FHIR is still in the early stages. Most of the resources that will be needed to create a useful system have not yet been designed and those that have are “proof of concept” and have not been vetted by their respective committees. It will be at least a year before there’s a version of FHIR available that’s suitable for any but the most “bleeding edge” of implementers. (Though trial versions of FHIR already exist for testing – see the website link at the end of this article.)

Once the specification is ready for use, the specific impact of FHIR is harder to gauge. It is unlikely that jurisdictions and implementations with a significant investment in existing standards (be that v2, CDA or v3 messaging) will drop existing functioning solutions to jump to FHIR. Therefore, early adoption is most likely to occur in green field areas, particularly mobile apps.

Further down the road, FHIR creates the promise of significant enhanced interoperability crossing the messaging, document and services paradigms. It also promises better interoperability across borders with a single wire format and consistent extensibility rules allowing common structures such as prescription, patient and lab result to share the same wire syntax regardless of jurisdiction. Unfortunately, FHIR does not solve the issues of terminology variation or varying legislative requirements driving the use of “must understand” extensions - so full out-of-the-box interoperability is still some way off.

## What does this mean for HL7 work groups?

FHIR moves away from the idea of “design by constraint” that was introduced with HL7 v3. While laudable in its intention to ensure semantic consistency across implementations, the result has often been standards that are too complex to easily implement or understand, and development work mired in committees attempting to satisfy every potential use-case. Instead FHIR aims to satisfy the common set of requirements and use a robust extensibility mechanism with a consistent wire format to manage the rest.

This represents a significant change for HL7. In both v2 and v3 specifications, the barrier was relatively low to modify the specification to introduce additional data elements. (Witness the PD1 segment ☺) Work groups will now need to push back more rigorously on requirements to ensure that only those elements that are truly core will be part of a specification.

It also represents a change in the perception of extensions. While extensibility is supported in HL7 v2, its use is frowned upon by implementers. Z-segments are opaque. No-one knows what the data inside them means unless they can contact the analyst at the site that originated the message. HL7 v3 messaging and CDA also support extensibility through the use of foreign namespaces, but that too is frowned upon because of the negative implications it has for schema validation and other processing. V3 and CDA extensibility is also limited to data elements that can be safely ignored by the receiver.

With FHIR, extensibility is a core part of the specification. Most wire instances will have at least a few extensions. It will be common for HL7 to define (and vet) extensions for common

## Timeline

FHIR is moving rapidly. First proposed at the Sept. 2011 working group meeting, FHIR will be going to informative ballot in July, 2012. First DSTU ballot will occur in the fall 2012 ballot cycle, with the hope of the first DSTU version of FHIR available by the end of 2013. Several work groups already have projects in place to develop FHIR resources, with others expected to come on this year and the first part of next year.

The initial focus for work groups will be defining the resources needed to cover their domain and, when necessary, extensions to cover “important” elements used by the domain, likely in an existing HL7 specification such as v2 or v3 messages or CDA implementation guide. Later, focus will shift to defining profiles of resources to meet specific interoperability use-cases.

## Licensing

The HL7 Board has recently begun to license a subset of HL7’s intellectual property for free. This includes HL7 Domain Analysis Models (DAMs) and HL7’s functional profiles. However, FHIR marks the first time that a core interoperability specification has been released under a free and open license. This license allows FHIR to be implemented, profiled and otherwise used without any requirement to pay membership or other licensing fees to HL7. The board has committed to this licensing approach until at least after the release of the first normative edition of FHIR, expected to be published in 3-4 years.

### Governance

Essential to FHIR’s success is the consistency with which FHIR principles are followed in the development of resources. For example, resources cannot overlap – a given clinical concept needs to fit in one and only one resource. As well, committees need to adhere to the rule of only including data elements that fit within the 80% - Those elements that 80% of implementations of the resource will support.

To ensure this consistency, the TSC has fashioned two governance bodies that will oversee resource allocation and development – the FHIR Governance Board (FGB) and the FHIR Management Group (FMG). Development and maintenance of the FHIR methodology and guidelines will remain the province of the Modeling and Methodology (MnM) work group.

<insert circle diagram showing relation of 3 bodies>

The FGB will be responsible for managing the authority to create resources – what resources will be created, what their scope delineation will be and what work group will be responsible for their content. It is also responsible for documenting and maintaining the principles under which FHIR will be managed and approving the guidelines created by MnM to ensure the quality of FHIR artifacts.

The FMG will be responsible for working directly with work groups developing resources and helping to facilitate cross-work group coordination and ensure consistency in the identification of what data elements are in the 80%. It will work to enforce the guidelines on resource quality developed by MnM and approved by the FGB.

### Next steps

The FHIR specification can be found here: <http://hl7.org>/fhir. Discussion about FHIR development takes place on both the FHIR HL7 wiki page: ???? and on the FHIR list server (categorized under the Technical Steering Committee section) on the HL87 list server page: ???. Take a look, provide your feedback, participate in the ballots, pass on the other information to others and play your part in this important initiative.

# Comparison of V2, V3, CDA and FHIR

The following table compares and contrasts the v2, v3 messaging, CDA and FHIR specifications, highlighting the things each does well and what each does poorly. It may help in an evaluation of the respective specifications and highlights some of the benefits of FHIR. The color-scheme is informal. Green indicates “close to ideal state”. Orange indicates “far from ideal state”. Yellow indicates “somewhere in between”.

| Characteristic | V2 | v3 messaging | CDA | FHIR |
| --- | --- | --- | --- | --- |
| *Capability* |  |  |  |  |
| Breadth of coverage | Administration and diagnostic-focused | Pretty much any healthcare scenario  you can imagine | Limited to “common”, patient-specific clinical scenarios | Pretty much any healthcare scenario  you can imagine |
| International Scope | Significant US-specific content | International | Mostly international (some popular templates US-centric) | International |
| Communication Paradigms | Messaging only | Messaging, some Services | Documents | Messaging, Documents, Services, REST |
| Support for complex scenarios | Limited data types and structures. Limited complexity handled in extensions | Robust support for complexity | Support for complexity in “in-scope” areas | “uncommon” complexity handled via extensions |
| Human readable content | Sometimes – by profile in NTE segment | Not usually | Yes | Yes, when desired |
| Extensibility | Yes (Z-segments) | Sort of (foreign namespace or special attribute) - discouraged | Sort of (foreign namespace) - discouraged | Yes (extensions section) – commonplace |
| Unknown extensions understandable? | No (unless you have analyst phone #) | Usually (if RIM naming followed) | Usually (if RIM naming followed) | Yes – via URL of extension |
| Out of the Box interoperability | Rarely. Interface engines required | Almost never at Int’l level – significant realm constraint needed | Yes – for human-to-human, simple metadata; templates required for system to system | Yes for REST if no “must understand” extensions invoked |
|  |  |  |  |  |
| *Architecture* |  |  |  |  |
| Object-oriented | No | Yes | Yes | Yes |
| Discrete, re-usable, context-independent components | Yes (Segments) | Sort of (CMETs) | Sort of (Entries) | Yes (Resources) |
| Robust semantics | No | Yes (not always modelled well). Intrinsic to presentation | Sort of (quality depends on template, some semantics inexpressible) | Yes (for resources), where possible for extensions) |
| *Wire format* |  |  |  |  |
| Modern / well supported wire format | Character delimited, some XML | XML (theoretical others) | XML | XML (JSON option) |
| Wire format backward/forward compatibility | Yes | No | Partial (within releases, not always between releases) | Yes |
| Human readable wire syntax? | Low – count vertical bars, guess based on data values | Low – bloated XML, formal names non-intuitive | Low – bloated XML, formal names non-intuitive | High |
| Size | 2304 pages (v2.7) | > 10k pages | 200 pages (+ ~400 of infrastructure) | ~1000 pages (estimate) |
| Learning curve | Moderate | Very high | High | Moderate |
| *Conformance* |  |  |  |  |
| Conformance profiling | As of 2.5, not commonly used | Conformance rules defined, generally handled via constrained constrained artifacts | Conformance rules defined. Templating only, typically captured as Word, though some formal capture, no formal declaration mechanism | Yes - profiles computer processable |
| Conformance declaration | Done using profiling mechanism | no formal declaration mechanism | no formal declaration mechanism | Yes – conformance statements computer processable |
|  |  |  |  |  |
| *Additional considerations* |  |  |  |  |
| Modeling expertise needed for design | None | High – key part of design | Moderate – needed for good template creation | Moderate – needed in parallel with resource and extension creation |
| Modeling expertise needed for implementation | None | Moderate – model is central in all documentation, non-fixed structural codes must be populated correctly | Moderate – model is central in all documentation, non-fixed structural codes must be populated correctly | None (though information available for those wanting it) |
| Standard creation tools | MS Word, validation / extraction via custom MS Access | Numerous custom applications, Windows-specific, custom generation tool | Same as v3 with manual editing of output of generation tool; additional custom tools for template authoring | Excel or Open Office, Enterprise Architect?, custom generation tool |
| Publishing approach | One spec, document form | Multiple interdependent specs, web-based | One main spec, few additional specs, web-based | One spec, document or web-based |
| Code generation | Some open source tools | Some open source tools | ??? | Part of published specification |

# What v3 issues does FHIR address?

The following table identifies a number of the issues that make implementing v3 specifications “hard”. For each, it identifies how much FHIR addresses the issue and provides descriptive comments about how it is addressed (or not)

| **Issue** | **Addressed?** | **Comment** |
| --- | --- | --- |
| Wire format hard to read | Mostly | Core content is easy to read. Extension content automatically convertible to be easy to read |
| Specifications hard to read/navigate | Mostly | Specification volume significantly reduced, focus is on “easy to read for implementers”, RIM modeling is suppressed |
| RIM experts hard to find | Mostly | RIM understanding no longer needed to read the spec or as critical path for resource or extension development. Still need RIM expert to do mappings, but can be more efficient and leverage fewer resources |
| Poor QA on v3 specifications | Partially | Smaller volume of elements (countable number of resources, ~80% reduction in data elements combined with stronger governance should allow better QA. As well, the requirement for instances and the expectation of committees to build (and thus validate) frequently should help too. Work Groups will be strongly discouraged from checking in content that fails validation.  Voluntary HL7 vetting process for extensions should help quality there too. |
| Non-interoperability across borders (project/geographic) | Partially | Wire format is the same everywhere – all countries, all domains. However, constraints can be different and some extensions may be “must understand” which can still interfere with interoperability. |
| Local variation of data elements | Partially | With FHIR, all extensions are sent in a consistent way that doesn’t break schemas. And you’re expected to accept unrecognized extensions. So long as an extension isn’t “must understand” you can create an app that populates all extensions everyone needs and send everywhere and it’ll work. |
| Local variation of constraints | No | If you make an element required, you’re not going to inter-operate with someone who has it as optional until they change their code. Same issue with minimum/maximum number of repetitions. |
| Local variation of terminology | No | If you send different codes than receiver requires, you’re not going to interoperate. |
| Terminologies are hard | No | FHIR doesn’t eliminate issues around complex terminology hierarchies, post coordination, etc. |
| Gaining consensus takes a long time | Tiny bit | In theory, a resource that’s at the 80% allows the consensus making to focus on extensions rather than on core, though there’ll still be a need to discuss constraints on core. And consensus is always hard. |
| Hard to make local changes quickly and push up | Partially | With FHIR, it becomes possible for a local project to throw together resources and extensions with little modeling expertise on a short timeline and still be reasonably aligned with a clear transition path to pan-Canadian equivalent that may take much longer to get resourced. Changes will be for RIM mappings (transparent to most implementers) and possible changes to some of the extensions. (manageable) |
| Content not shareable across paradigms | Yes | Content is represented the same way in documents, messages, services and REST. (Documents **may** supplement with additional organization/rendering structures.) |
| Creating generic templates (usable across multiple specifications) is hard | Mostly | With FHIR, there’s a single base model for each clinical concept – the resource. It has fixed element names. Profiles written against that resource can apply against all instances, regardless of jurisdiction, discipline, domain, etc. Still an issue for templates intended to apply across resources, but countable number of resources makes it manageable. |
| Proliferation of templates/profiles | Not really | FHIR makes it easy to create and roll out template (profile) registries. But proliferation is a governance issue, not a technical issue. |
| DCM consensus is hard | Tiny bit | FHIR - due to extensible, discrete and x-paradigm nature provides good foundation for DCMs. However, consensus process and issue of “boiling the ocean” (extremely large number of needed DCMs) |
| Interoperable discrete data capture impacts application design | No | Interoperability means capturing discrete data in standardized ways using standardized vocabulary. There’s no way for this to avoid impacting (and often significantly impacting) application design. |
| Interoperability affects workflow | No | If the workflow isn’t standardized, your applications won’t interoperate. FHIR supports many workflow paradigms. It doesn’t simplify agreeing on one. |