RDF as a Universal Healthcare Exchange Language: Realistically Achieving Semantic Interoperability

David Booth, Ph.D., KnowMED, Inc.

Latest version of this document: http://dbooth.org/2013/rdf-as-univ/rdf-as-univ.pdf
Slides: http://dbooth.org/2013/rdf-as-univ/slides.pdf

Abstract

Electronic healthcare information is currently represented in a bewildering variety of incompatible data formats, models and vocabularies. To improve healthcare effectiveness and efficiency, healthcare computer systems need to be able to exchange electronic healthcare information in a machine processable form that enables semantic interoperability between systems. This paper explains how a *universal healthcare exchange language* can be realistically adopted in order to achieve semantic interoperability between healthcare systems. It explains why RDF is the best available candidate for such a language, and it outlines work needed to prove viability on a national or international scale.

Introduction

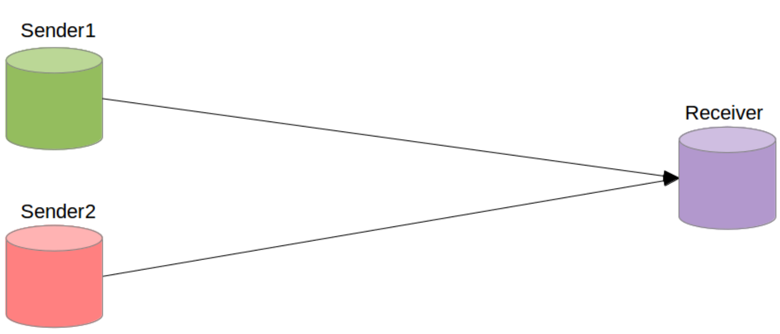
Imagine a world *in which all healthcare systems speak the same language with the same meanings covering all healthcare.* True semantic interoperability: what would it be like?

- *Better treatment*, as doctors could more easily obtain an automatically integrated view of a patient's condition and history.
- *Better research*, as researchers could more easily combine and analyze data from many sources.
- *Lower cost*, as efficiency would be improved.

Unfortunately, electronic heathcare information systems today are a Tower of Babel, using hundreds of different and incompatible data formats, models and vocabularies, thus inhibiting semantic interoperability. The President's Council of Advisors on Science and Technology (PCAST) highlighted the importance of this problem and called for a *universal exchange language*: "PCAST has also concluded that to achieve these objectives it is crucial that the Federal Government facilitate the nationwide adoption of a universal exchange language for healthcare information".[1]

What is semantic interoperability?

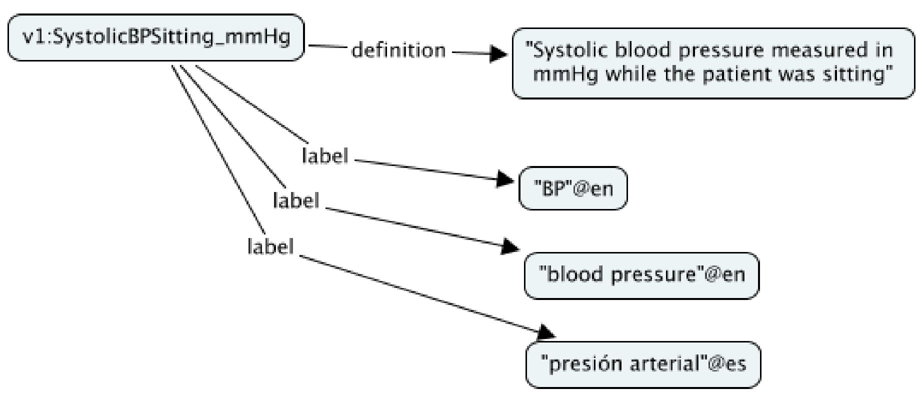
Suppose a patient is being treated by a healthcare provider (the information Receiver), and that patient's medical records are requested from two other healthcare providers (Sender1 and Sender2) in order to automatically obtain an integrated view of the patient's condition and history. If semantic interoperability were achieved, the Receiver system, given appropriate authorization and routing information, could: (a) obtain the patient's medical records from Sender1 and Sender2; (b) combine those records into an integrated patient record; and (c) extract any needed information from the result -- all automatically.



To achieve this automatically (without human labor), the information must be expressed in a **machine processable** format that computer systems can properly interpret, both efficiently and reliably. A machine processable format is a structured format that enables a computer system to properly "understand" the data, i.e., to make meaningful use of it. This is in contrast with human oriented information formats, such as narrative text or scanned documents. Human oriented information formats can be efficiently and reliably generated from machine processable formats, but not vice versa.

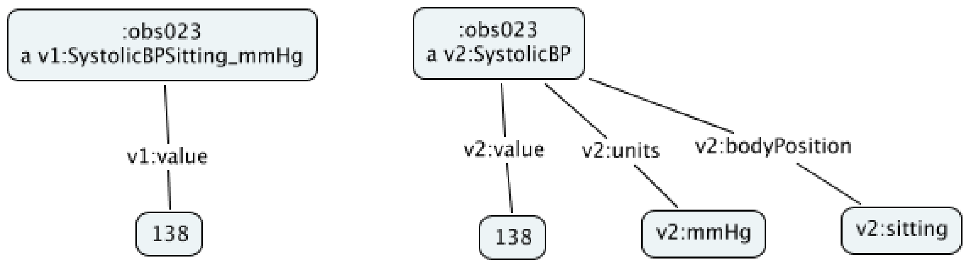
Even if a data format is understood, for the Receiver system to properly interpret the received information it contains, the information must also be expressed in a **controlled vocabulary** that the Receiver understands. However, the use of standard medical terms is not enough to ensure accurate interpretation, because the same term may be understood differently by different parties -- sometimes in subtly different ways. For example, to assess a patient's health risk factors, a data element may capture information about whether the patient smokes. But the term "current smoker" may have different meanings in different contexts.[2] To ensure accurate interpretation of captured data, terms in a controlled vocabulary must have agreed-upon *definitions*.

On the other hand, sometimes different natural language terms mean the same thing, i.e., they have the same definition. For example, "blood pressure", "BP" and "presión arterial" (Spanish) may all refer to the same concept of blood pressure. Thus, to facilitate machine processing, avoid ambiguity and enable meaningful information display to human users, it is helpful to represent each term in a controlled vocabulary as an **unambiguous concept** consisting of: a unique *identifier*; a *definition*; and one or more human-oriented *display labels*. For example if v1 is a controlled vocabulary then v1:SystolicBPSitting_mmHg may be a globally unique identifier for a particular concept of a blood pressure measurement, having an associated definition and multiple display labels.



The Receiver system does not necessarily need to directly understand each controlled vocabulary term it encounters, but if it encounters a term that it does not understand, then there should be a standard, automatable algorithm to enable the Receiver to obtain an accurate definition of the term. Whenever possible, the definition should be provided in both a machine-processable form and a human-oriented form that relates the term to other terms that are already known to the system, so that the system can bootstrap its understanding. For example, in **Linked Data**[3], each term is unambiguously identified by a URI[4]. If a system does not already understand a term, the system can *dereference* the term's URI to obtain the term's definition.

Another complexity in semantic interoperability is that different parties may represent information using **different data models**. For example, a systolic blood pressure measurement might be encoded either in a "pre-coordinated" style that also indicates the body position and the units of measure, or in a "post-coordinated" style that breaks the information into atomic pieces, as shown below. [Thanks to Stanley Huff for this example.] Both representations carry the same information.

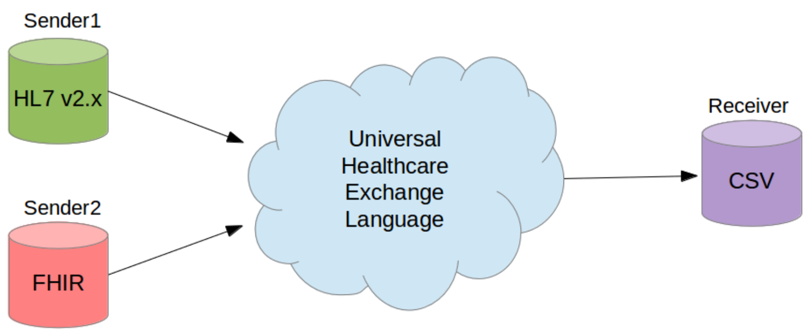


Fortunately, **metadata** concepts in a vocabulary can **signal the expected data model**. Such metadata should be included with the instance data to enable a receiving system to properly interpret the data. For example, if an observation is recorded as a v1:SystolicBPSitting_mmHG measurement (using vocabulary v1), then the units and body position may already be implied by that concept's definition. Alternatively, if the observation is recorded as a v2:SystolicBP measurement (using vocabulary v2), then the use of that identifier (v2:SystolicBP) signals the use of a data model in which the value, units and body position are explicitly indicated as associated data elements, using terms v2:value, v2:units and v2:bodyPosition. In summary, the vocabulary defines a set of concepts along with their associated data models, and the data models indicate the relationships between the concepts. Such a vocabulary is sometimes called an *ontology*. Finally, because a concept can signal what data model is used, the problem of standardizing the data models boils down to the problem of standardizing the concepts: if the concepts are standardized, the data models implicitly become standardized.

Metadata can also carry **data provenance** information, for tracing data origins and stewardship, and it can carry access control information for enforcing **privacy and security** requirements. While privacy and security enforcement is critically important in any healthcare system, the question of semantic interoperability only arises *after* any privacy and security questions have been addressed and proper authorization has been granted. Hence this paper focuses only on semantic interoperability, with the understanding that any system must first enforce all privacy and security requirements.

The role of a universal healthcare exchange language

In theory, semantic interoperability could be achieved by converting all healthcare systems to internally use one standard format and vocabulary. However, this option is neither politically feasible nor advisable, both because of the enormous transition investment that it would require and because it would stymie innovation. Hence **we will assume that healthcare systems will continue to internally use whatever data formats and vocabularies they choose**, and examine what is needed to achieve semantic interoperability in the *exchange* of healthcare information between systems. To illustrate, let us assume that Sender1 uses one format (HL7 v2.x[5]), Sender2 uses another (FHIR[6]), and Receiver uses a third (CSV, comma-separated-values[7]).



To achieve semantic interoperability, the data from Sender1 and Sender2 must be somehow transformed into the format and vocabulary that the Receiver can understand. The purpose of a *universal healthcare exchange language* is to simplify this transformation by enabling healthcare information to be transformed to and from a common intermediate language. Although the use of an intermediate language means that two transformations must be performed during each information exchange instead of one – source to intermediate language, and intermediate language to destination – it dramatically reduces the implementation complexity by reducing the number of kinds of transformation that are needed. Instead of needing $(n-1)*(n-1)$ transformations only n are needed, where n is the number of distinct kinds of sender/receiver languages. If healthcare systems eventually choose to use this common language internally, all the better, but that is not a requirement for achieving semantic interoperability.

Why is semantic interoperability so difficult?

If a universal healthcare exchange language can be standardized, and all parties exchange healthcare information using the same standard data models and vocabularies, then semantic interoperability can be achieved. This sounds simple, but it is deceptive. It would be easy enough to standardize a sufficiently flexible *syntactic* framework for such a language. But standardizing the *semantics* -- the data meaning -- is far more difficult. Data meaning is determined by the data models and vocabularies that are used in the data. There are three main reasons why these are so hard to standardize:

- Medicine is very complex, involving many thousands of interrelated concepts and many overlapping areas of expertise.
- As the size and complexity of a standardization task grows -- and the number of committee members grows -- the rate of progress diminishes toward zero.
- Medical science and technology are continually changing, requiring new concepts all the time. It is a moving target.

In short, it is not feasible to stop the world until we can standardize all of the data models and vocabularies needed for a universal healthcare exchange language to achieve full semantic interoperability. Instead, we need an approach that acknowledges and accepts the dynamic nature of the problem.

Key requirements for a universal healthcare exchange language

Because of the above challenges, a viable universal healthcare exchange language must meet several key requirements:

- The language must accommodate the **continual incorporation of new or revised data models and vocabularies**.
- The language must support the use of **existing and future healthcare information standards**, to protect investments and reap the benefits of standardization whenever possible.
- The language must support **decentralized innovation**, to avoid committee bottlenecks and enable new concepts to be used before they have been standardized.

Furthermore, to support the graceful adoption of new data models and vocabularies:

- The language should enable new or revised data models and vocabularies to be **semantically linked** to existing data models and vocabularies.
- The language should enable **authoritative definitions** of new concepts to be obtained

automatically, so that when a system encounters a new term, the system can properly understand it.

The *best available candidate* to meet these requirements is RDF.

What is RDF?

Resource Description Framework (RDF)[8] is an information representation language. An international standard, RDF is qualitatively different from other information representation languages currently common in healthcare, due to its *schema promiscuity* (explained below), its basis in web standards and its emphasis on semantics. RDF acts as a unifying substrate for healthcare information, allowing any number of vocabularies and data models to be semantically connected and used together. How this works, and why this is important, will be explained in the remaining sections. RDF does not and cannot solve the semantic interoperability problem by itself -- no technology can, as there are important social factors at play in addition to technical factors -- but RDF can be a crucial component as a universal healthcare exchange language.

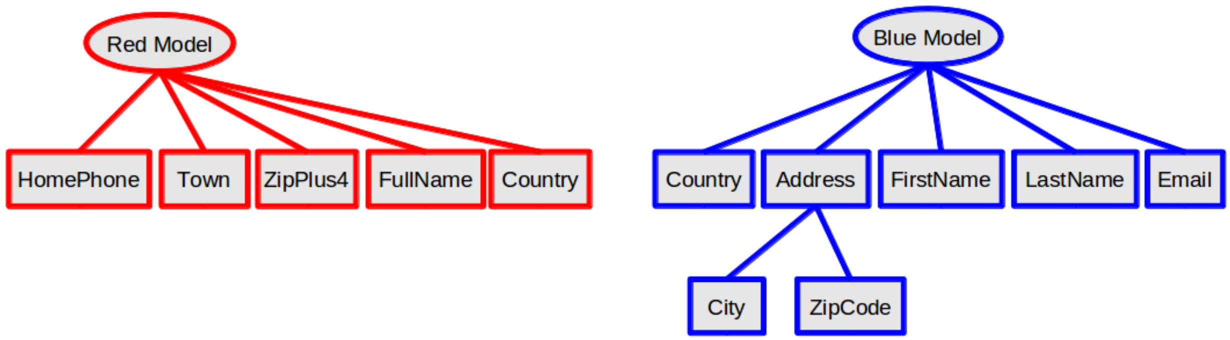
URIs as unique identifiers

URIs are the basis of the World Wide Web, and can be used both as web page locators and as globally unique identifiers. RDF uses URIs as globally unique identifiers. Any concept used in healthcare can be given a URI as an identifier, including concepts in vocabularies, procedures, medications, conditions, diagnoses, people, organizations, etc. The use of URIs enables orderly decentralized allocation of identifiers, and, if Linked Data principles are adopted, it means that an identifier can be conveniently associated with its definition: a URI can be dereferenced to a web page to find its definition.

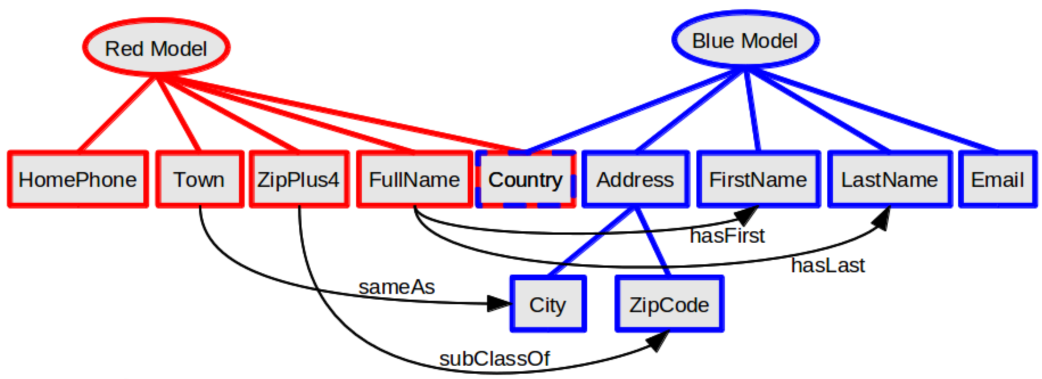
Schema promiscuity

The most important characteristic of RDF that makes it well suited to be a universal healthcare exchange language is what we may call *schema promiscuity*. In contrast with most common information representation languages such as XML, RDF allows data expressed in different data models or *schemas* to peacefully coexist within the same datasets, semantically interlinked. Relationships between concepts in different models can be expressed in RDF just as relationships within a model are expressed. Instead of requiring one model to incorporate all others, RDF allows any number of models to be in use at the same time. Different applications can have different views of the same data, and the addition of a new view does not impact existing views.

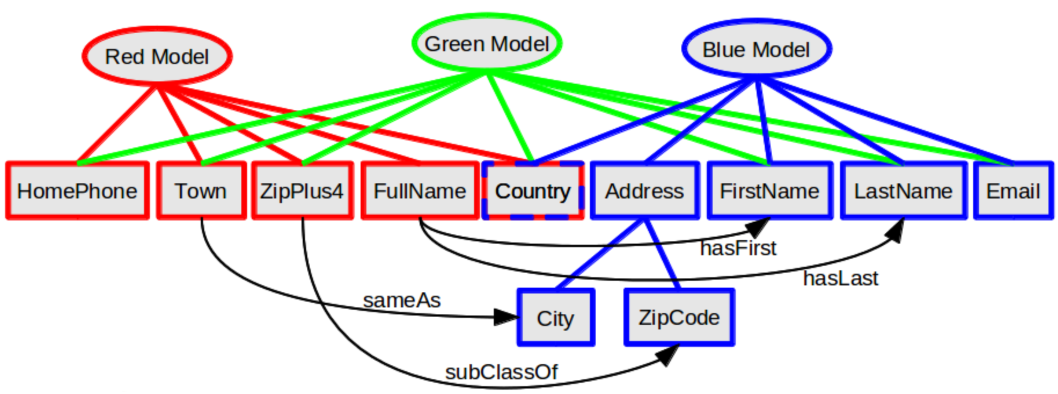
For example, two customer address models – Red and Blue – may have been developed independently for different applications, each one representing similar information in different ways.



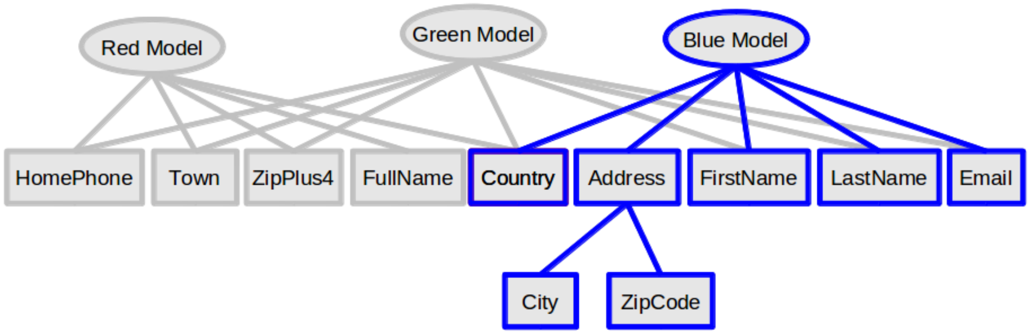
Later, these two models are used together in the same data, and a **bridge model** is used to link them together.



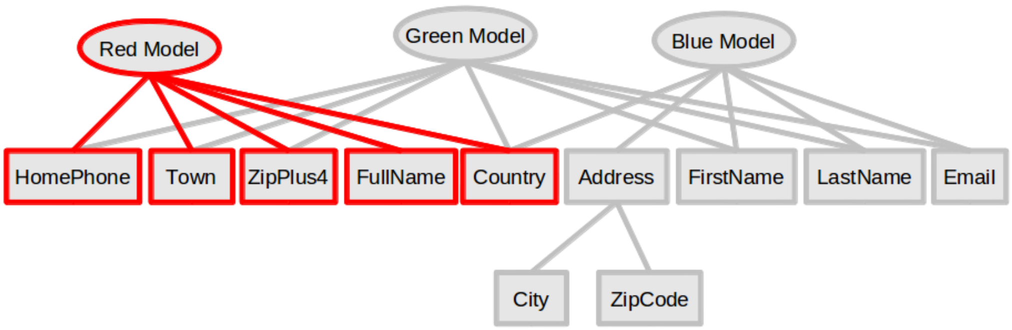
The bridge model adds more RDF relationships (shown in black) to semantically link concepts in the Red and Blue models that are related. Concepts that were already used in common between the two models, such as Country (as determined by use of the same URI), join automatically with no need for a linking relationship. A Green application may later make use of elements from both the Red and Blue models to form its own Green model.



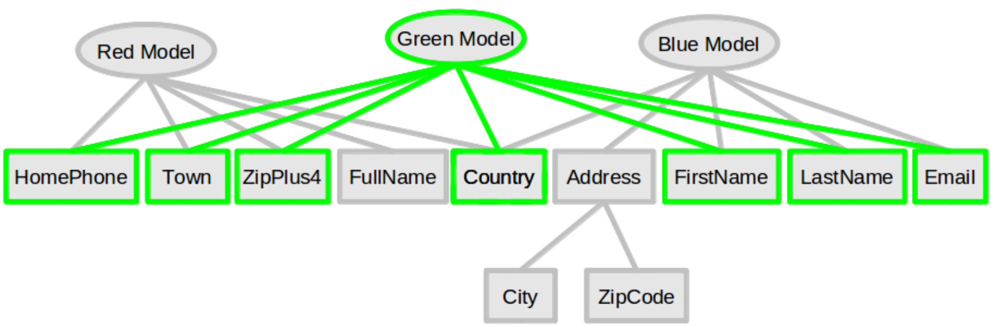
Even though all of these models now coexist, and are semantically linked, each application still retains its own preferred view of the data. The addition of other data models does not break existing applications. The Blue application still sees the Blue model,



the Red application still sees the Red model,



and the Green application sees the Green model, which happens to make use of portions of the Red and Blue models.



Furthermore, by exploiting the semantic links between models, information that originated in one model may be automatically transformed to a second model for use by applications that expect the second model, provided that bridge transformations are available. Some transformations are easy. For example, a 9-digit zip code from the Red model (ZipPlus4) is already a kind of Blue model zip code (ZipCode), since the additional four digits are optional in the Blue model zip code: every ZipPlus4 is a ZipCode (but not vice versa). Other transformations may depend on multiple data elements. For example, to create a 9-digit ZipPlus4 postal code, information from ZipCode, Country, City and Address must be used. Transformations can also draw upon information from multiple models at once. In some cases a transformation may require the addition of external information, if the transformation requires more information than existing models have captured.

This ability to simultaneously accommodate any number of semantically linked models is crucial because it allows new data models and vocabularies to be continually incorporated, without breaking existing software. It also allows information to be represented at multiple levels of granularity simultaneously, which is important because different applications have different needs.

Emphasis on semantics

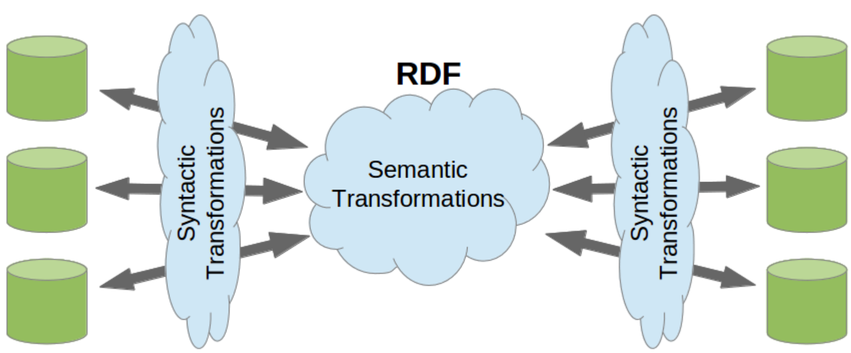
RDF is **syntax independent**: RDF data expresses an abstract model of the information content that is independent of its serialization. There are multiple serialization formats (or syntaxes) for RDF. The same exact information content (or RDF abstract model) can be serialized in different ways. Four of the most popular serializations are N-Triples[9], which serializes RDF as a very simple list of subject-verb-object *triples*; Turtle[10] which is a more compact form that is easier for humans to read and write; JSON-LD[11], which is a more recent JSON-based format; and RDF/XML[12], which is an older XML-based format that is now mostly only used for historical reasons, because other formats are now easier to understand and use.

Being syntax independent, RDF places an emphasis on the meaning of the information rather than on its syntactic representation. This also means that any data format with a suitable mapping to RDF can be used as RDF – regardless of whether it was originally designed to be an RDF serialization. This is significant, because it means that existing document formats can be treated as specialized RDF formats and used along with other RDF content. Existing standard formats do not need to be discarded and reinvented in RDF. Instead, a transformation can be defined from the existing format to the RDF abstract model.

Data transformations

RDF does not by itself give us semantic interoperability. Rather, it acts as a common language substrate for enabling semantic interoperability. If data originates in one form, but is required in a different form, there is no way around the need to transform the data from one form to the other. For the moment we will ignore details of *where* and such transformations will be performed -- they could be done by senders, receivers, intermediaries, or some combination thereof -- and focus only on *what* needs to be done and how it can be achieved.

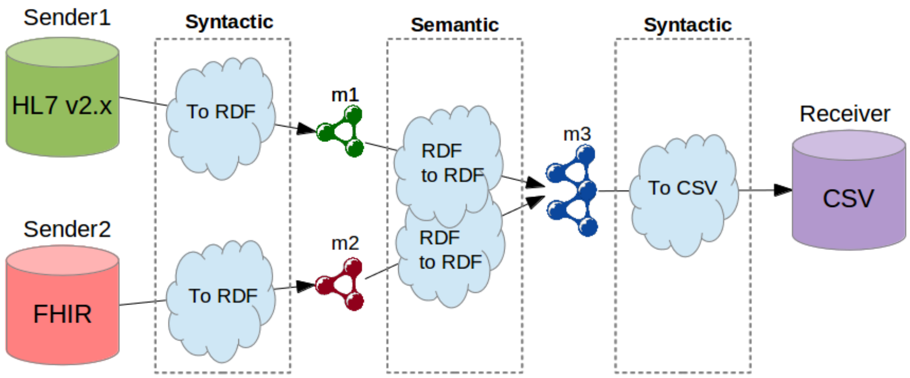
To simplify the task of creating these data transformations is it helpful to conceptually subdivide them into *syntactic transformations* and *semantic transformations*. The syntactic transformations are used to



convert between existing data formats and RDF. The semantic transformations are used to convert between models and vocabularies within RDF. This conceptual division of responsibilities allows the semantic transformations to be potentially reusable across different data formats, and it allows the semantic transformations to be as simple as possible, because they do not need to deal with the idiosyncrasies of the data formats.

Syntactic transformations

To illustrate the syntactic transformations in our example of Sender1, Sender2 and Receiver, HL7 v2.x data from Sender1 must be transformed into RDF. This transformation is not intended to yield the final RDF that will be needed, but transforms the data into an RDF version of Sender1's **native model**, m1, which corresponds directly with the information content of the HL7 v2.x data, without attempting to map to a particular destination model. Model m1 uses the same vocabulary as the original HL7 v2.x data and directly reflects its existing information model (ignoring superficial syntactic details). Similarly, FHIR data from Sender2 is transformed to an RDF version of Sender2's native model, m2.



There are two reasons for transforming first to this native model instead of attempting to transform all the way to the destination model or even to a common intermediate model. The first is that it simplifies the transformation: it can be implemented without knowledge of any other models. This also allows these syntactic transformations to be performed using generic tools. For example, the W3C has defined a generic Direct Mapping from relational databases to RDF.[13] The result of the W3C Direct Mapping is RDF that directly reflects the information model of the relational database to which it is applied.

The second reason for transforming first to the native model is that it avoids tightly coupling the syntactic transformation to any particular destination model, and this allows the transformation to be more reusable. If a new destination model is desired, the same syntactic transformation can be used, and only a new semantic transformation will be needed. It also better insulates the semantic transformations from changes to the syntactic transformation that are the result of changes to the original data format.

As an example, suppose Sender1 sends an HL7 v2.x message looking something like the following (simplified for clarity):

```
OBX|1|CE|3727-0^BPsystolic, sitting||120||mmHg|
```

A syntactic transformation may convert this to RDF expressed in model m1 like the following (written in Turtle, with namespaces omitted for brevity):

```
d1:obs042 a m1:PatientObservation ;
  m1:code "3727-0" ;
  m1:description "BPsystolic, sitting" ;
  m1:value 120 ;
  m1:units "mmHg" .
```

Similarly, Sender2 may send a FHIR message looking something like the following (simplified for clarity):

```
<Observation xmlns="http://hl7.org/fhir">
   <system value="http://loinc.org/"/>
```

```
    <code value="8580-6"/>
    <display value="Systolic BP"/>
    <value value="107"/>
    <units value="mm[Hg]"/>
</Observation>
```

A syntactic transformation may convert this to the following RDF:

```
d2:obs-091 a m2:Observation ;
  m2:system "http://loinc.org/" ;
  m2:code "8580-6" ;
  m2:display "Systolic BP" ;
  m2:value 107 ;
  m2:units "mm[Hg]" .
```

Syntactic transformations are also used in the opposite direction: to serialize to a particular required data format after semantic transformations have been performed. In the diagram above, the transformation would be from a third information model m3 (discussed next) to a comma-separated-values (CSV) format required by Receiver.

Semantic transformations

Although the syntactic transformations have converted Sender1 and Sender2's data into RDF, the data is expressed in models m1 and m2 that Receiver will not understand, because Receiver understands only model m3. Therefore, semantic transformations are performed from RDF to RDF to transform from source models and vocabularies to destination models and vocabularies. The purpose is to achieve **semantic alignment**: to express all of the information in the same, desired information model and vocabularies – in this case m3 – so that the information can be meaningfully combined and understood by the Receiver.

There are many ways these semantic transformations can be performed. Here is an example of one technique, which uses a SPARQL[14] query as a transformation rule to convert from Sender1 model m1 to model m3 (namespaces omitted for brevity):

```
# Transform m1 to m3
CONSTRUCT {
  ?observation a m3:Observation ;
    a m3:BP_systolic ;
    m3:value ?value ;
    m3:units m3:mmHg ;
    m3:position m3:sitting .  }
WHERE {
  ?observation a m1:PatientObservation ;
    m1:code "3727-0" ;
    m1:value ?value ;
    m1:units "mmHg" . }
```

Here is the transformation from model m2 to model m3:

```
# Transform m2 to m3
CONSTRUCT {
```

```
    ?observation a m3:Observation ;
      a m3:BP_systolic ;
      m3:value ?value ;
      m3:units m3:mmHg . }
  WHERE {
    ?observation a m2:Observation ;
      m2:system "http://loinc.org/" ;
      m2:code "8580-6" ;
      m2:value ?value ;
      m2:units "mm[Hg]" . }
```

How can a system automatically know what semantic transformations to apply? Metadata, also expressed in RDF and carried with the data, can indicate what data models and vocabularies are used. By also knowing what data models and vocabularies the Receiver expects, a system can automatically deduce what semantic transformations are needed. For example it may determine that it needs to transform from m1 to m3, or from m2 to m3. It could then lookup the appropriate transformations in a catalog and apply them. Such transformations might not be performed in a single step, but may involve a short series of transformations forming a **transformation path** from source model(s) to destination model.

Incorporating non-standard concepts

RDF permits both standard and non-standard vocabularies to be used and intermixed, using the techniques described above. This provides the best of both worlds: standard vocabularies can be used to whatever extent they are available, thus simplifying transformation processes and enabling semantic interoperability, while non-standard or emerging vocabularies can be used whenever they are needed, and can be linked with standard vocabularies. This allows the adoption of standard vocabularies to proceed gracefully at whatever pace is possible.

One may wonder why it is important to accommodate vocabularies (or individual concepts) that have not yet been standardized. If a concept has not been standardized, what good will it do to include it in the transmitted data? How will anyone know how to interpret it? The answer is that the adoption of new concepts does not happen all at once by all parties. Some parties will be able to make use of a new concept even before it is standardized. Indeed, they have a business incentive to do so. And if Linked Data principles are used, then each concept's definition can be easily obtained by dereferencing the concept's URI, this making it easy to bootstrap the use of new concepts. The inclusion of non-standard concepts also smooths the path toward their eventual standardization, by allowing interested parties to gain experience with them.

However, the ability to accommodate non-standard concepts is both a blessing and a curse. The downside is that it could allow healthcare providers to use non-standard concepts *even when standard concepts are available*, which would impede semantic interoperability rather than enabling it. And unfortunately, since healthcare providers in the USA have no natural business incentive to make their data understandable by others (especially their competitors), incentives must be provided in other ways to ensure that standards are used whenever possible. Therefore, **carrot and/or stick incentives must be enacted to encourage the use of designated standards**.

The problem of proprietary vocabularies

High-quality healthcare vocabularies cost money to create and maintain. Hence, some have restrictive licensing requirements that prevent them from being freely used. This creates a barrier to their use. To best enable semantic interoperability, this barrier should be reduced or eliminated.

The aspect of this barrier that is most important to reduce or eliminate is the barrier to use (or *read*) and understand data that was expressed using a proprietary vocabulary. While one party may choose to gain the benefits of using a proprietary vocabulary when capturing (or *writing*) its healthcare data, another party that needs to receive and understand data from the first party should not be forced to sign or pay for a license to that vocabulary in order to properly interpret and use the received data.

Policymakers can reduce these barriers by encouraging or requiring the use of vocabularies that are free and open for use in reading – and ideally also writing – healthcare information.

Recipe for semantic interoperability

In the exchange of healthcare information, semantic interoperability cannot be realistically achieved all at once in a "big bang" fashion. But it can be realistically achieved on a progressive basis in which the scope of semantic interoperability is as large as possible and continually increases as more concepts become standardized. This section outlines key principles for achieving this goal.

1. Exchanged healthcare information should be **machine-processable**, as structured data, whenever possible. *This is important to enable useful, automated machine processing.* Narrative or other unstructured information should also be included as an adjunct to convey additional information that cannot be adequately conveyed as structured data.
2. Exchanged healthcare information should be in an **RDF-enabled format**, either: (a) in a standard RDF format directly; or (b) in a format that has a standard mapping to RDF. *This is important to simplify information transformations, by adopting a common language substrate.*
3. Designated **standard vocabularies** should be used whenever possible, i.e., for all concepts that are defined in designated standard vocabularies. *This is important to simplify semantic interoperability.*
4. The set of designated standard vocabularies should be **continually expanded and updated**. *This is important to continuously increase the scope of semantic interoperability.*
5. ***All* requested information** should be provided in an RDF-enabled format – not only those concepts that are available in designated standard vocabularies. For concepts that are not (yet) defined in designated standard vocabularies, best available vocabularies should be used. *This is important to enable the smooth evolutionary adoption of new concepts and to ensure that all potentially useful information is available to authorized requesters.*
6. Existing standard healthcare vocabularies, data models and exchange languages should be leveraged by defining **standard mappings to RDF**, and any new standards should have RDF representations. *This is important to preserve investments in existing and future standards.*
7. Exchanged healthcare information should be **self-describing**, using Linked Data principles, so that each concept URI is de-referenceable to its definition. *This is important to ensure consistency of interpretation and to efficiently bootstrap the adoption of new concepts.*
8. Healthcare information should be exchanged using **free and open vocabularies**. *This is important to prevent business and legal barriers from impeding the beneficial exchange and use*

of healthcare information. Note that this would not prevent healthcare providers from using proprietary vocabularies internally.

9. Adequate **incentives for semantic interoperability** should be enacted, to ensure adherence to the above principles. *This is important to overcome the lack of natural business incentive to achieve semantic interoperability in the healthcare industry.*

Proving feasibility

Many who have worked with RDF and related semantic web technologies believe that this approach to achieving semantic interoperability is viable.[15] All of the features of this approach have been applied individually in healthcare, the life sciences or other areas, but the totality of this approach has not been proven to work on the scale needed for nationwide (or worldwide) adoption. To demonstrate viability on a nationwide scale, work should be undertaken to do the following:

- Build a working, networked system – a reference implementation – that demonstrates this approach on a concrete end-to-end use case involving at least three parties – two senders and one receiver – in which healthcare information about the same patient is received from both senders and usefully combined and used by the receiver.
- Demonstrate the feasibility of all important features of this approach, including:
  - syntactic transformations
  - semantic transformations
  - selecting and applying semantic transformations
  - incorporating new vocabularies and deprecating old vocabularies, showing how these changes affect the selection and application of semantic transformations
  - hosting concept definitions
  - privacy and security adherence
- Design and run stress tests that simulate the adoption of this approach on a nationwide scale, to identify important scaling issues.
- Recommended a set of conventions that would facilitate adoption, such as conventions for identifying semantic transformations, for requesting RDF healthcare information, etc.

Conclusion

The successful adoption of a universal healthcare exchange language is more difficult than it may seem, but it is both feasible and worthwhile. The best available candidate is RDF, primarily because it enables multiple data models to be used together and semantically linked. The viability of this approach for achieving semantic interoperability on a nationwide scale should be proven by building, demonstrating and stress-testing a working reference implementation.

References

1. President's Council of Advisors on Science and Technology: Realizing the Full Potential of Health Information Technology to Improve Healthcare for Americans: The Path Forward.

Executive Office of the President, December 2010. http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-health-it-report.pdf (Accessed 25-Aug-2013)

2. Heather Ryan, Angela Trosclair and Joe Gfroerer: Adult Current Smoking: Differences in Definitions and Prevalence Estimates – NHIS and NSDUH, 2008. Journal of Environmental and Public Health, Volume 2012 (2012), Article ID 918368, 2012. http://dx.doi.org/10.1155/2012/918368 or http://www.hindawi.com/journals/jeph/2012/918368/ (Accessed 25-Aug-2013)
3. W3C: Linked Data. http://www.w3.org/standards/semanticweb/data (Accessed 25-Aug-2013)
4. T. Berners-Lee, R. Fielding, L. Masinter: Uniform Resource Identifier (URI): Generic Syntax. IETF Request For Comments (RFC) 3986, January 2005. http://www.w3.org/standards/semanticweb/data (Accessed 25-Aug-2013)
5. Health Level Seven International: HL7 Standards – Master Grid. http://www.hl7.org/implement/standards/product_matrix.cfm?ref=nav (Accessed 25-Aug-2013)
6. Health Level Seven International: Fast Health Interoperability Resources (FHIR) v0.11. http://www.hl7.org/implement/standards/fhir/fhir-book.htm (Accessed 25-Aug-2013)
7. Wikipedia: Comma-separated values. http://en.wikipedia.org/wiki/Comma-separated_values (Accessed 25-Aug-2013)
8. Graham Klyne, Jeremy J. Carroll, Brian McBride: Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-concepts/ (Accessed 25-Aug-2013)
9. Gavin Carothers and David Beckett: N-Triples, A line-based syntax for an RDF graph. W3C Working Group Note 09 April 2013. http://www.w3.org/TR/n-triples/ (Accessed 25-Aug-2013)
10. Eric Prud'hommeaux, Gavin Carothers, David Beckett, Tim Berners-Lee: Turtle, Terse RDF Triple Language. W3C Candidate Recommendation 19 February 2013. http://www.w3.org/TR/turtle/ (Accessed 25-Aug-2013)
11. Manu Sporny, Gregg Kellogg and Markus Lanthaler: JSON-LD 1.0, A JSON-based Serialization for Linked Data. W3C Last Call Working Draft 11 April 2013. http://www.w3.org/TR/json-ld/ (Accessed 25-Aug-2013)
12. Dave Beckett and Brian McBride: RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-syntax-grammar/ (Accessed 25-Aug-2013)
13. Marcelo Arenas, Alexandre Bertails, Eric Prud'hommeaux, Juan Sequeda: A Direct Mapping of Relational Data to RDF. W3C Recommendation 27 September 2012. http://www.w3.org/TR/rdb-direct-mapping/ (Accessed 25-Aug-2013)
14. Steve Harris, Andy Seaborne, Eric Prud'hommeaux: SPARQL 1.1 Query Language. W3C Recommendation 21 march 2013. http://www.w3.org/TR/sparql11-query/ (Accessed 25-Aug-2013)
15. David Booth, Charlie Mead, Michel Dumontier, Tracy Allison Altman, Rafael Richards, et al: Yosemite Manifesto on RDF as a Universal Healthcare Exchange Language. Position statement from the 2013 Workshop on RDF as a Universal Healthcare Exchange Language, San Francisco. http://yosemitemanifesto.org/ (Accessed 25-Aug-2013)

Change Log
25-Aug-2013 Completed initial draft.
3-Aug-2013: Initial draft.