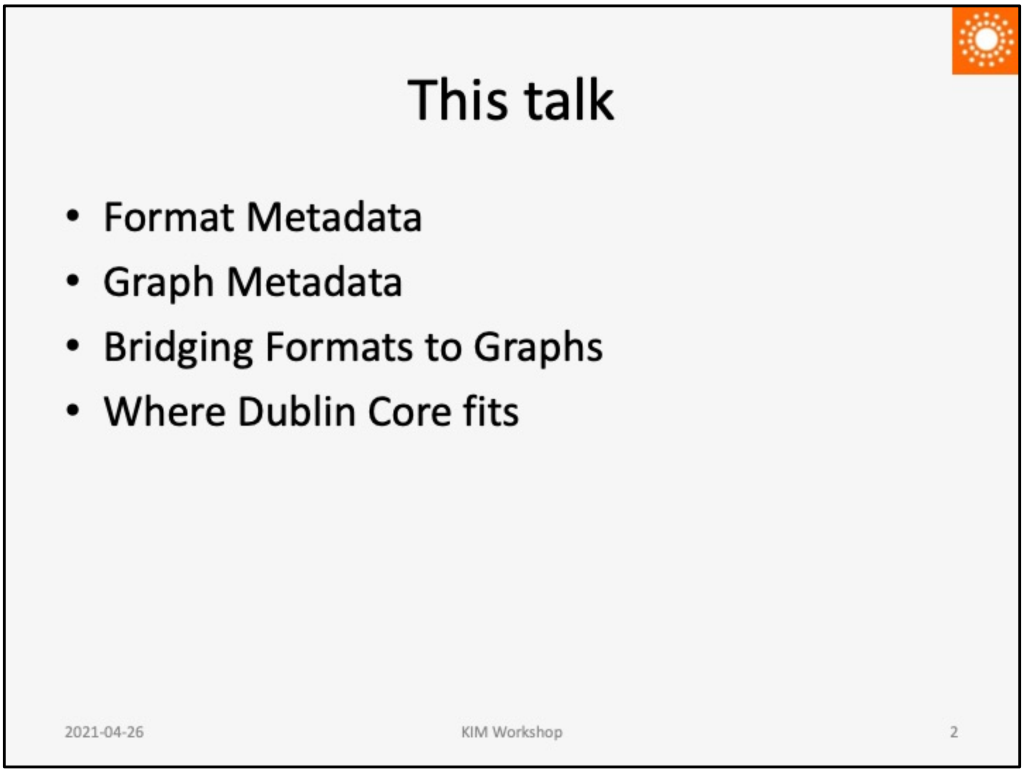
Beyond “Simple and “Qualified”: Dublin Core today

Tom Baker

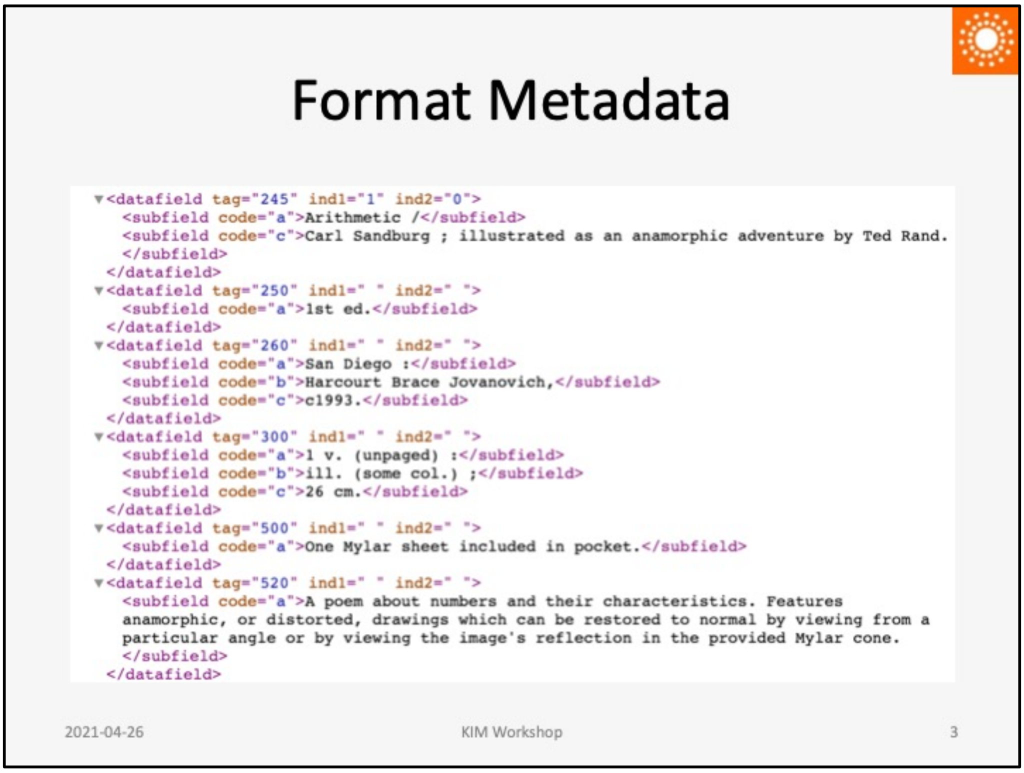
KIM Workshop, Germany

26 April 2021



I want to start today with the contrast between what I call Format Metadata and Graph Metadata.

I will then describe some technologies and models that are bridging these two approaches and comment on where Dublin Core fits in the picture.

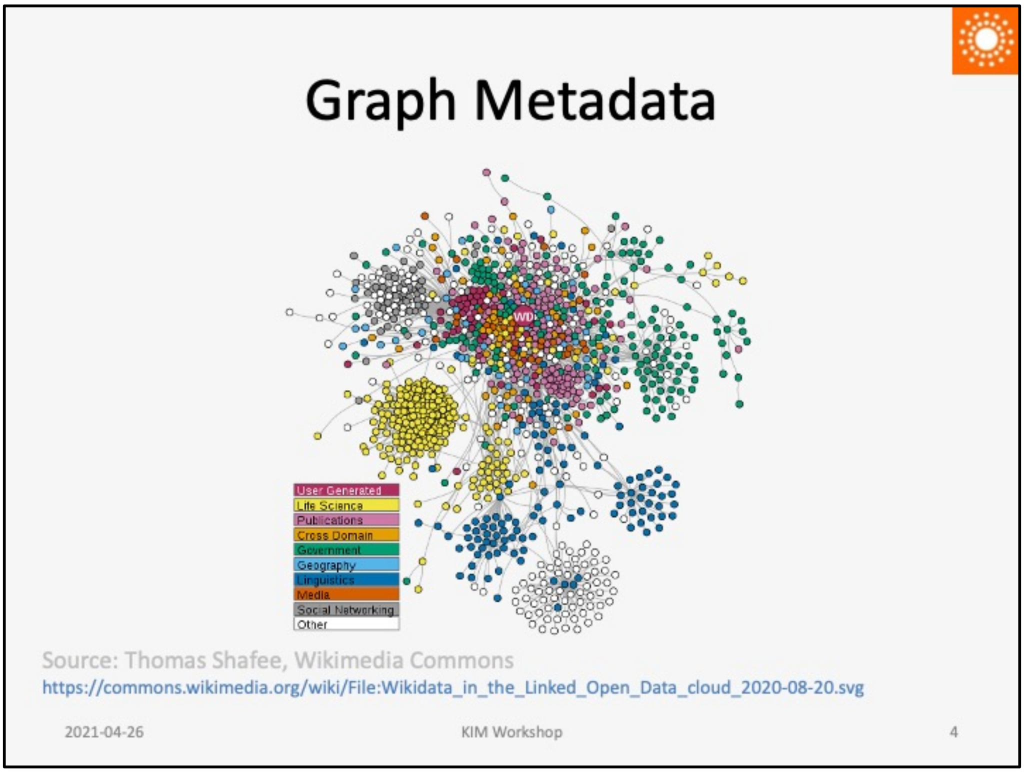


In traditional IT systems, "metadata" is understood to mean the fixed data structures found in databases or in record formats, also known as exchange formats, such as MARC records or XML Schema documents.

Let’s call this Format Metadata.

Format Metadata is designed for a specific application context -- a database, an institute, a company, or a network, even if that network is as large as the library world.

There is typically no strong requirement for the formats to be understood outside of that context.



When the Web arrived in 1990s, it became possible to access data from many sources and from anywhere.

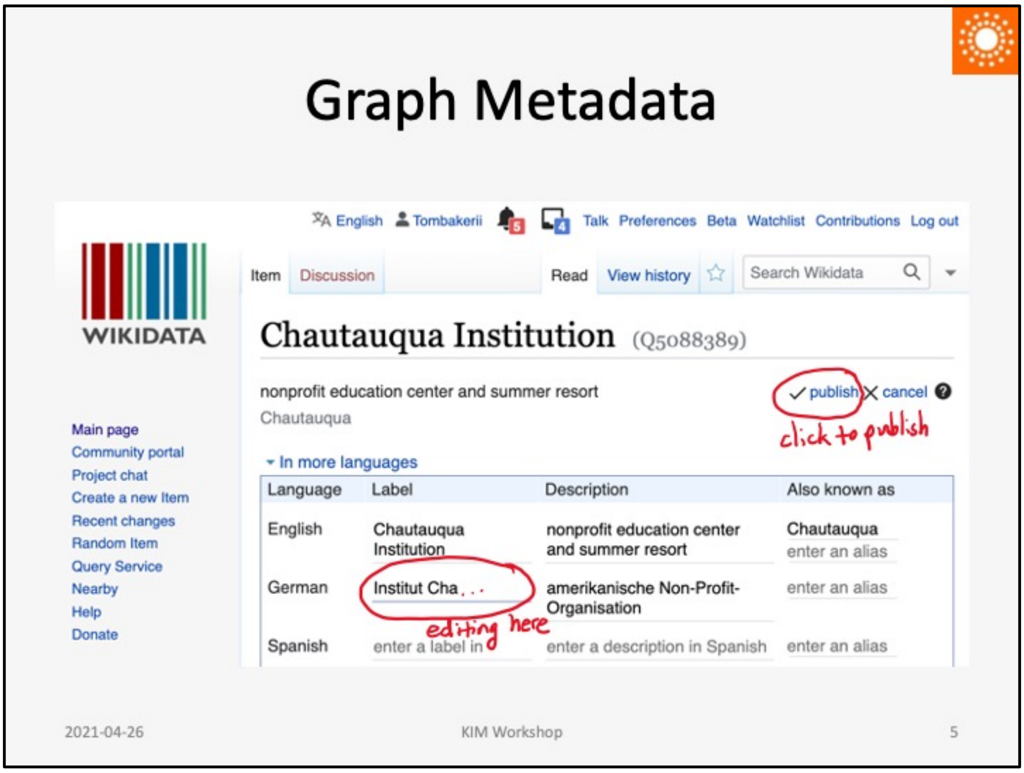
People wanted to mash up the data in new and different ways, but this was hard to do when the data was locked into different formats.

In response, W3C developed a generic model for expressing metadata as sets of individual statements, each meaningful on its own, linked into extensible structures called graphs.

Let’s call this Graph Metadata.

In theory, there is no limit to how large a graph can grow. Wikidata currently holds data on more than 93 million things.

This visualization shows how Wikidata connects data ranging from Government Information and Social Media to Linguistics and the Life Sciences.



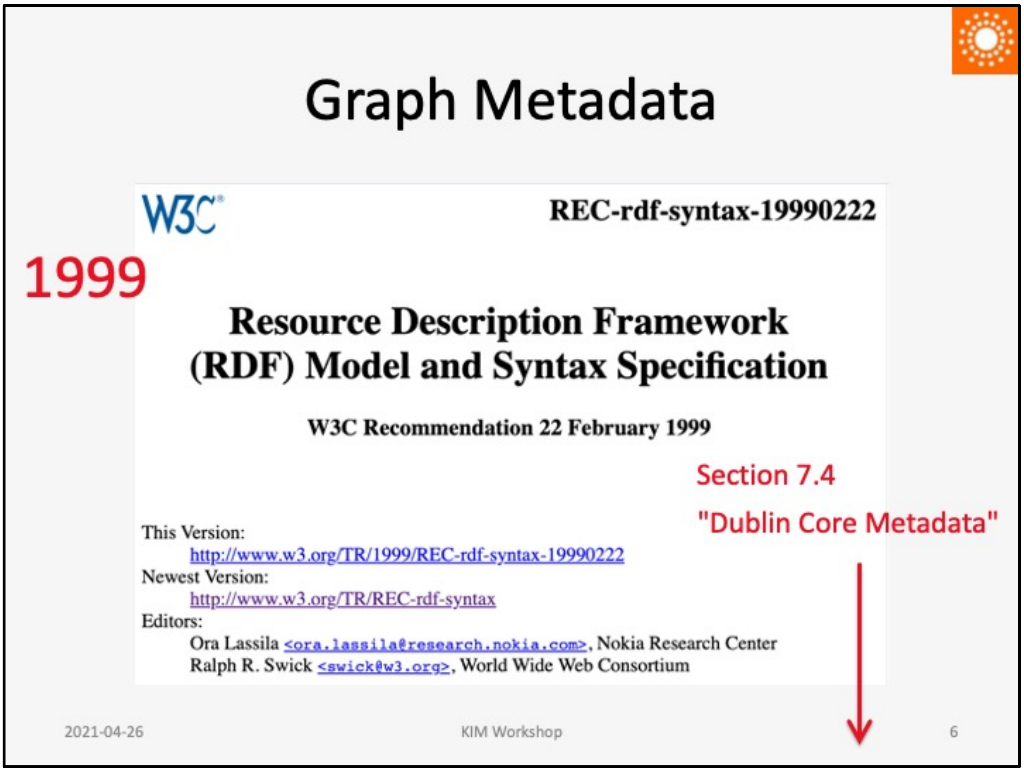
Wikidata is, in fact, an interesting case, as it shows the autonomy of individual statements at its most extreme.

Anyone in the world can edit Wikidata, adding statements here or there, and with no requirement to supply, or to follow, any sort of schema.

This screenshot shows me adding the German name for Chautauqua Institution.

Schemas are typically created in Wikidata after the fact, when data providers seek agreement among themselves on how to describe specific types of thing: for example, the properties and classes one might use to describe a hospital.

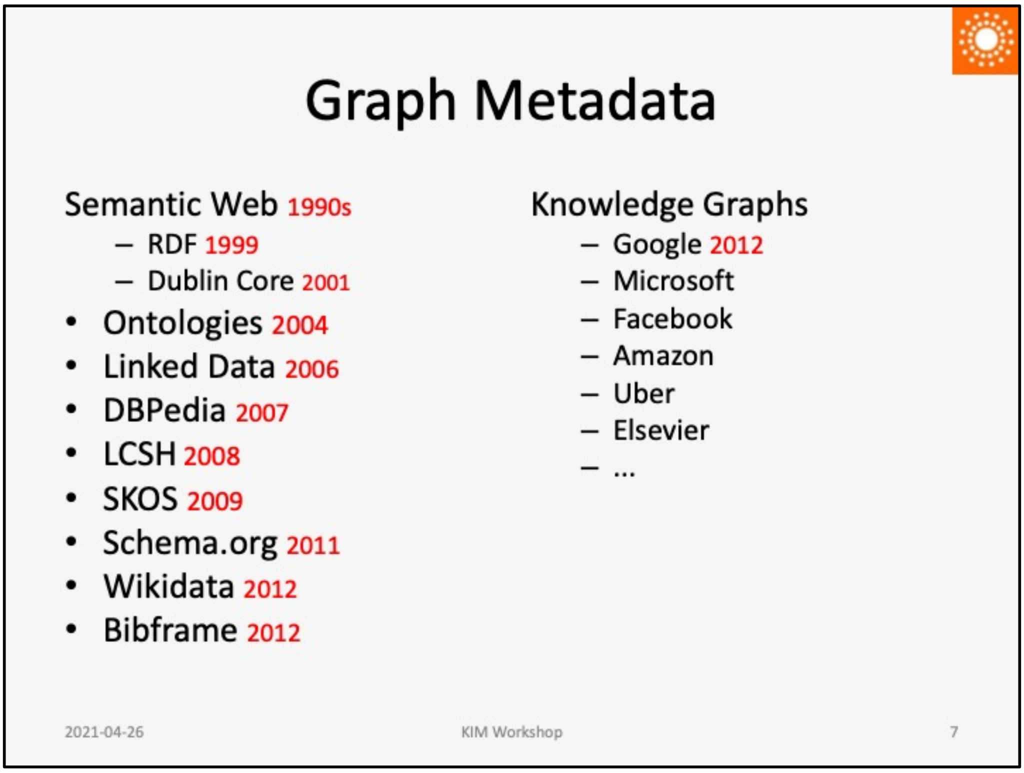
This agreement can be captured in schemas, which are also published in Wikidata. More about these later.



The idea of Graph Metadata first emerged in the 1990s.

A generic model – Resource Description Framework, or RDF – was developed to support the vision of a Semantic Web.

Dublin Core was in fact one of the very first vocabularies for RDF.

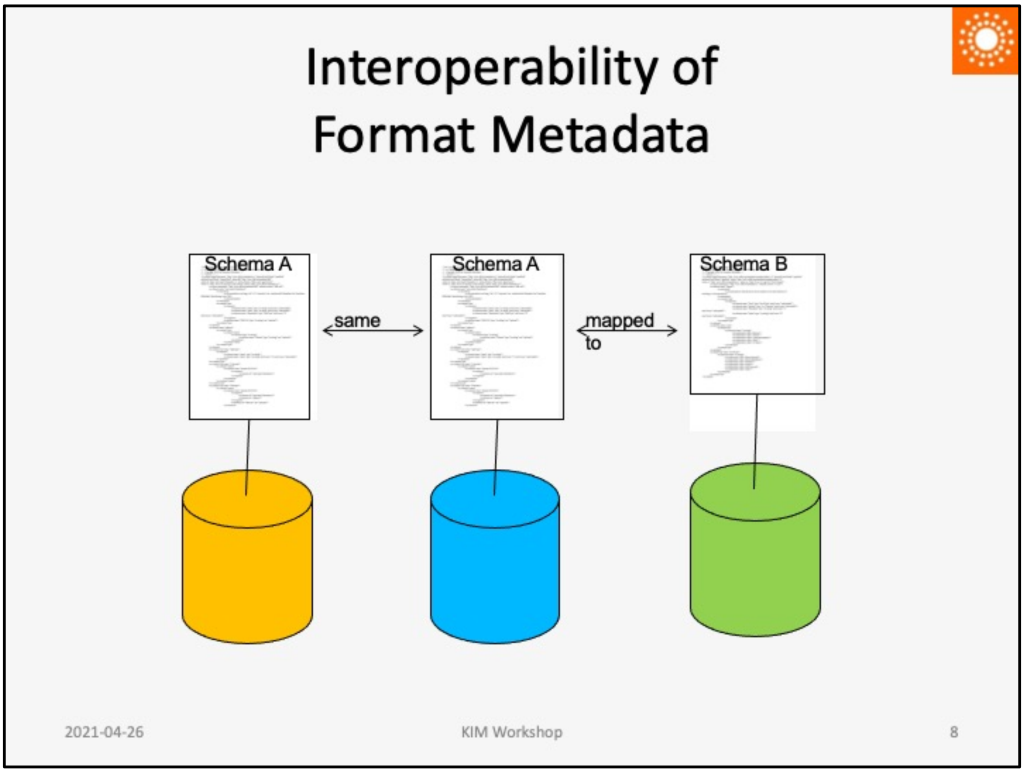


This was followed in the next decade by Ontologies, DBPedia, and Linked Data.

The publication of Library of Congress Subject Headings in SKOS was followed by the adoption of RDF in the library world. Next came Schema.org, Wikidata, Bibframe, and Google's Knowledge Graph.

Corporations such as Microsoft, Facebook, Amazon, Uber, and Elsevier turned knowledge graphs into key assets for their businesses.

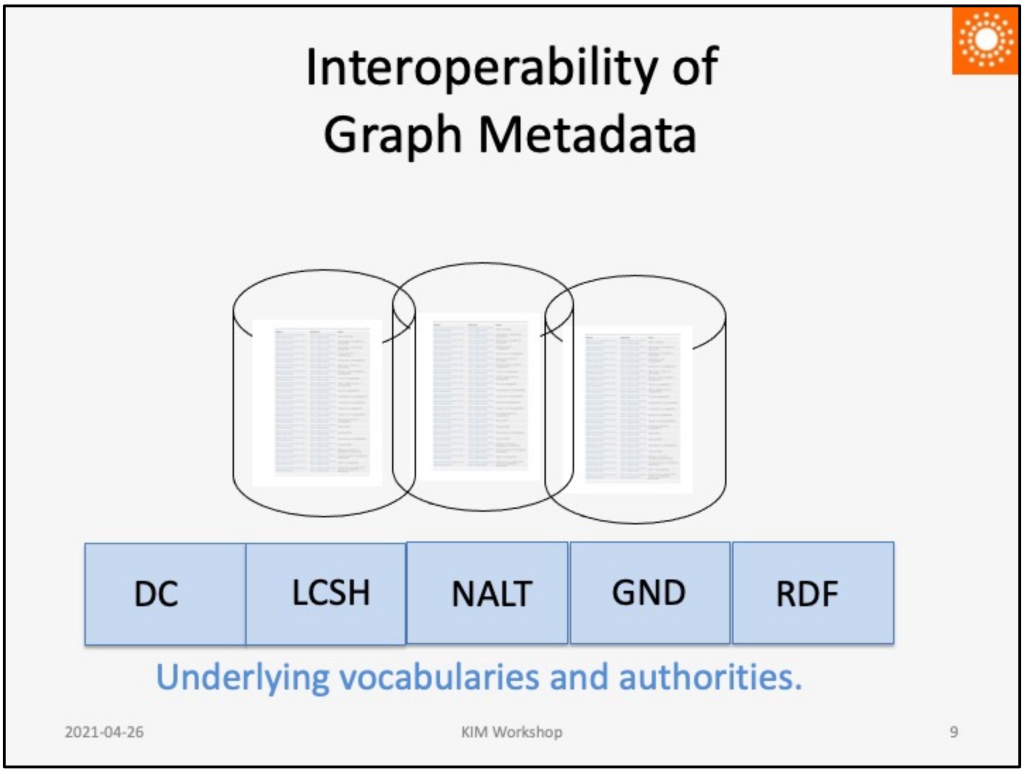
Corporate knowledge graphs may not all be based on RDF, or on open RDF vocabularies, but they share with RDF the idea of statements aggregated into graphs.



In Format Metadata, interoperability is achieved by ensuring that everyone uses exactly the same format, with the same constraints and the same tables or element structures.

In reality, of course, getting everyone to use exactly the same format is no easier than getting everyone to speak the same language.

Partial interoperability is achieved with "crosswalks" that map equivalent elements between different formats.

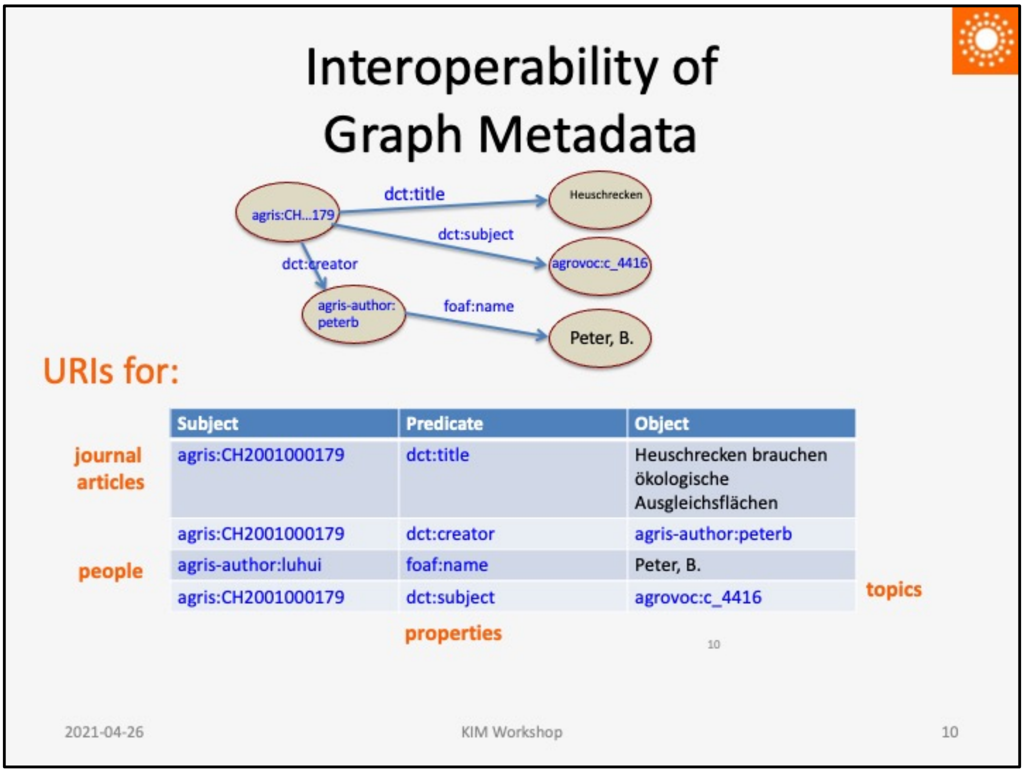


In Graph Metadata, interoperability is based in part on shared vocabularies.

RDF vocabularies such as DCMI Metadata Terms define properties and classes for describing things.

SKOS concept schemes defines hierarchies of topics for specific domains, such as AGROVOC or NALT, two thesauri of terminology in agriculture.

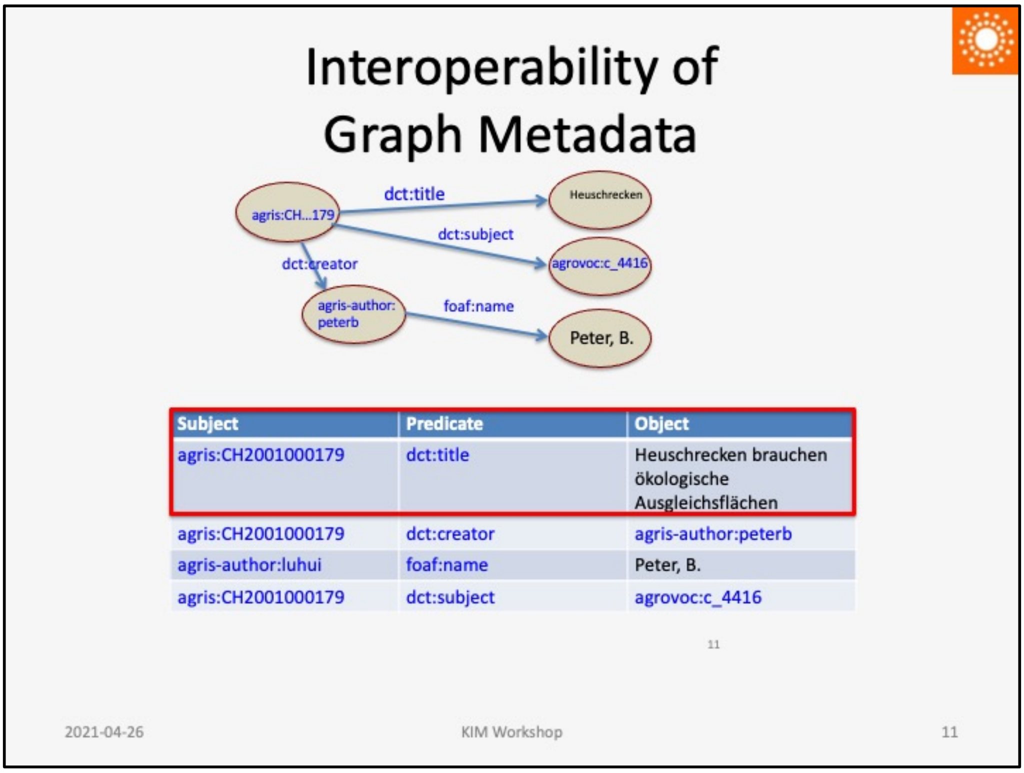
Note that RDF itself is an RDF vocabulary!



RDF vocabularies are composed of global Web identifiers, also known as URIs or IRIs.

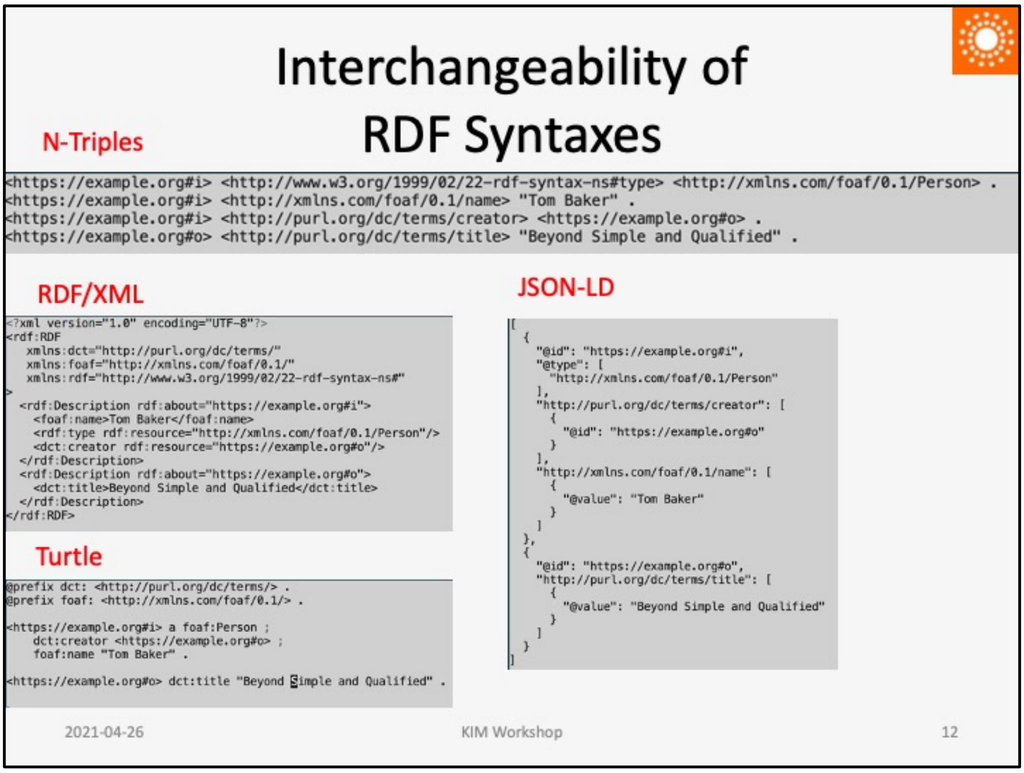
These URIs identify anything that can be named, such as specific people, journal articles, places, or topics. URIs can also identify properties of things, such as birthdate, publisher, or geolocation coordinates.

Because these vocabularies are typically published on the Web, in RDF, the Web in effect provides the language of graph metadata with its dictionaries.



Statements in this metadata language are based on a simple, three-part model of subject, predicate, and object called a "triple".

A statement reads like a simple sentence. This one says: "Journal Article so-and-so has the title 'Heuschrecken brauchen oekologische Ausgleichsflaechen'".



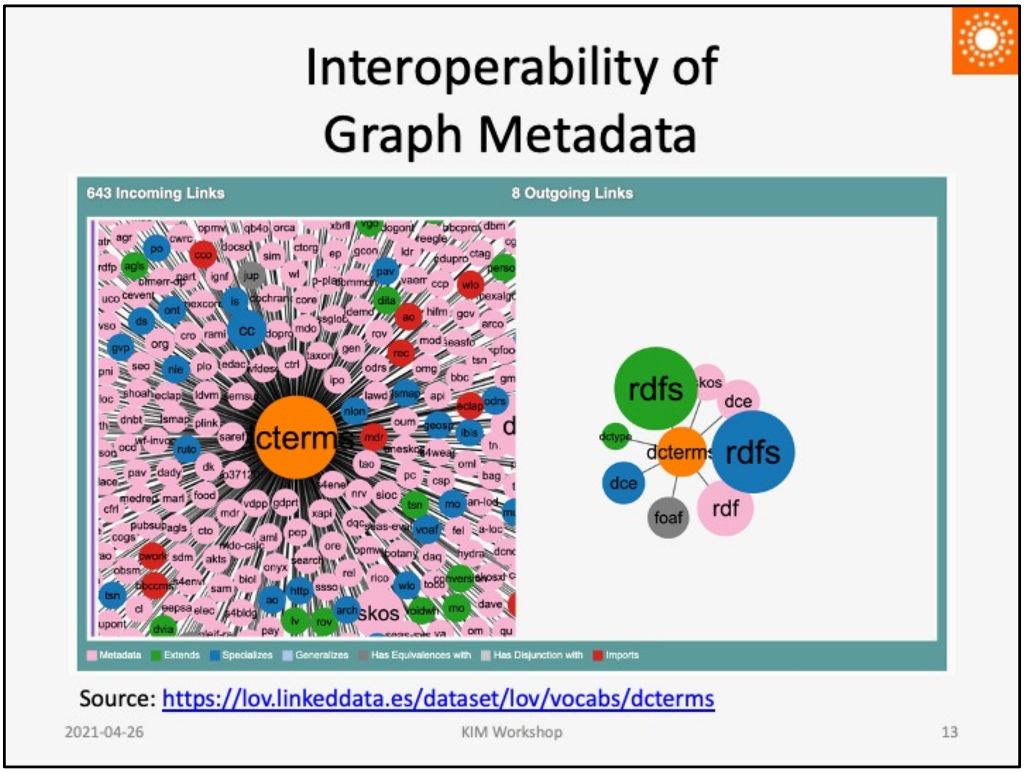
It is important to understand that the RDF triple is a model, and the model can be serialized in any one of several completely interchangeable ways.

Here is a simple RDF graph, with just three triples, serialized in four alternative syntaxes:

- N-Triples, the most straightforward, easy to process,
- Turtle, the easiest to read,
- RDF/XML, the original syntax from 1999, widely used but hard to read and widely disliked,
- JSON-LD, great for applications that read JSON and preserves sequential order.

Any one of these serializations can be converted into any of the others.

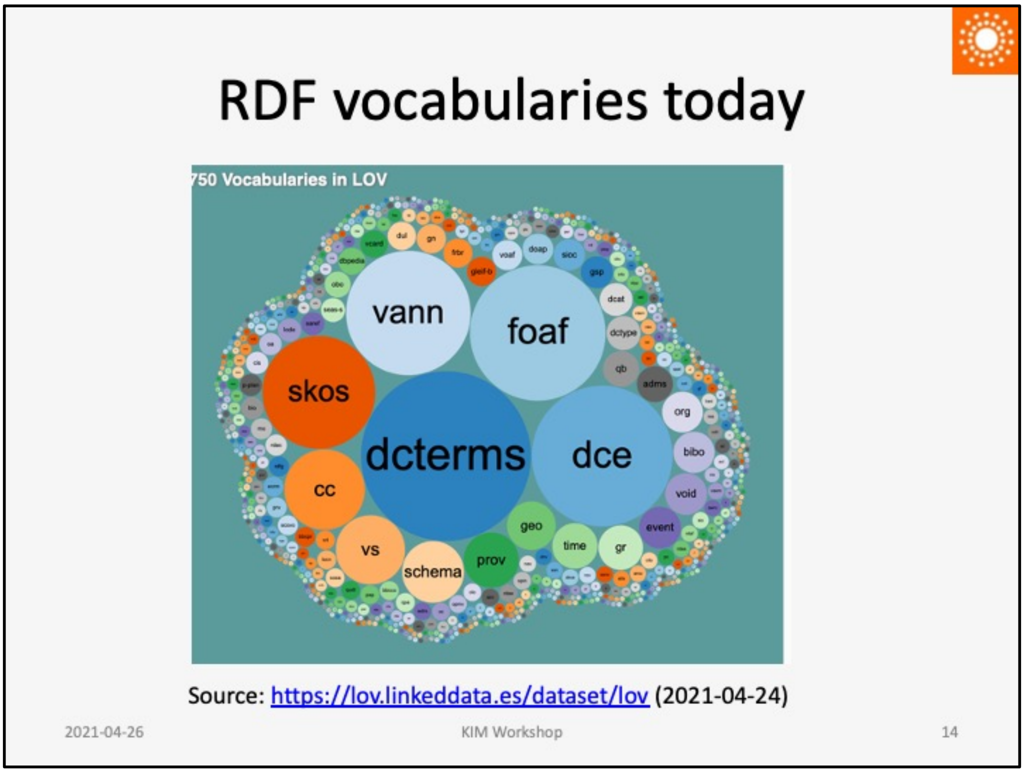
For developers with special requirements, there are even more syntaxes – not just these three.



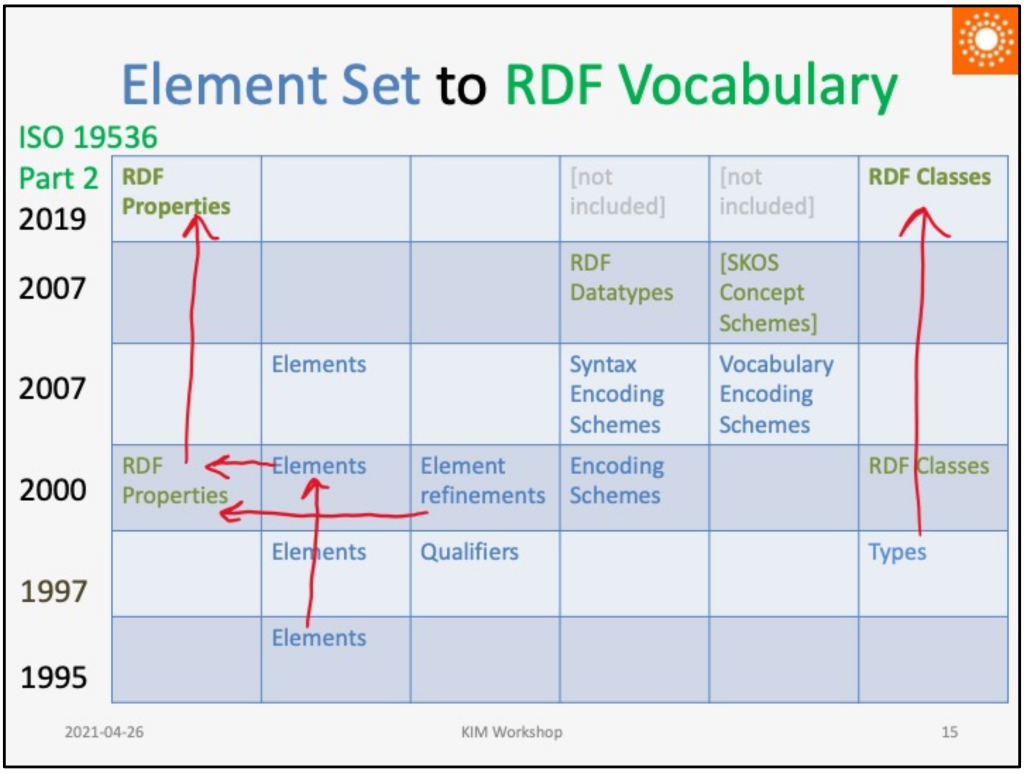
Because RDF also makes it easy to map equivalent properties and authority URIs from different sources, the interoperability of metadata does not depend on using _exactly_ the same vocabularies and authorities.

it can be enough to use vocabularies and authorities that have been mapped to each other.

Dublin Core has been mapped to alot of other vocabularies.



DCMI vocabularies – here, dcterms, dce, and dctype – are among the vocabularies most frequently used in Linked Data.



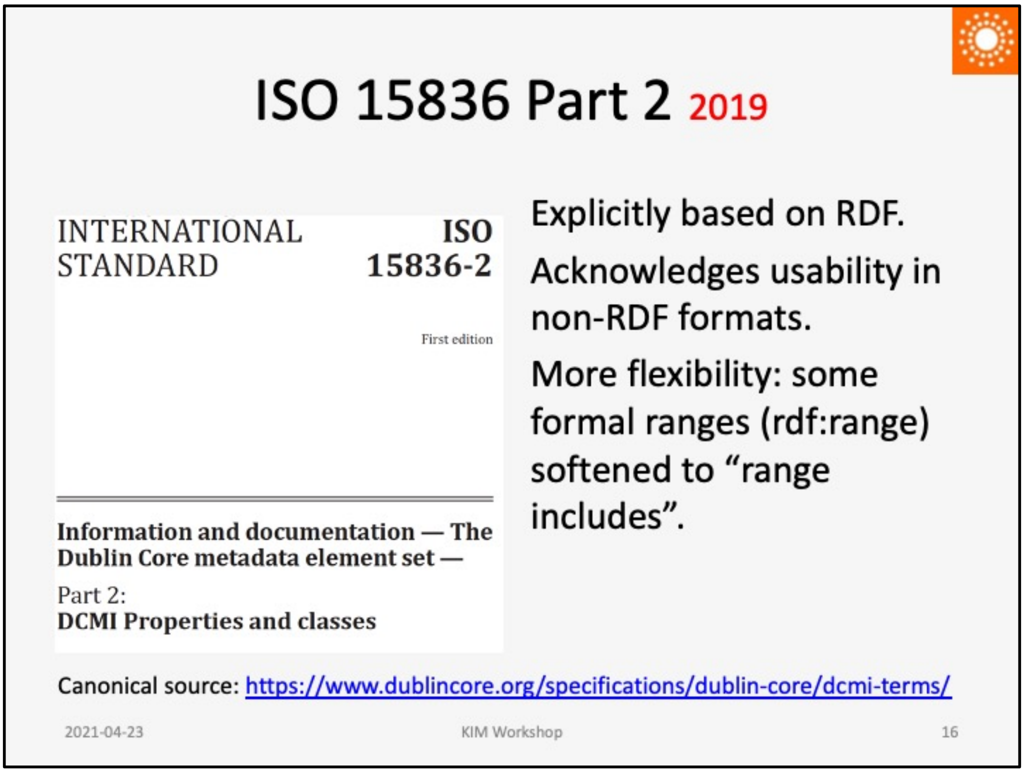
One sees the evolution of Dublin Core from Format Metadata to Graph Metadata in its terminology.

In the early years, the Dublin Core Metadata Element Set was defined as a simple format of fifteen elements.Then qualifiers were added to support the use of elements with more precision.

From 2000 to 2007, Dublin Core was increasingly seen as an RDF vocabulary.

The most frequently-used terms in that vocabulary are Properties and Classes.

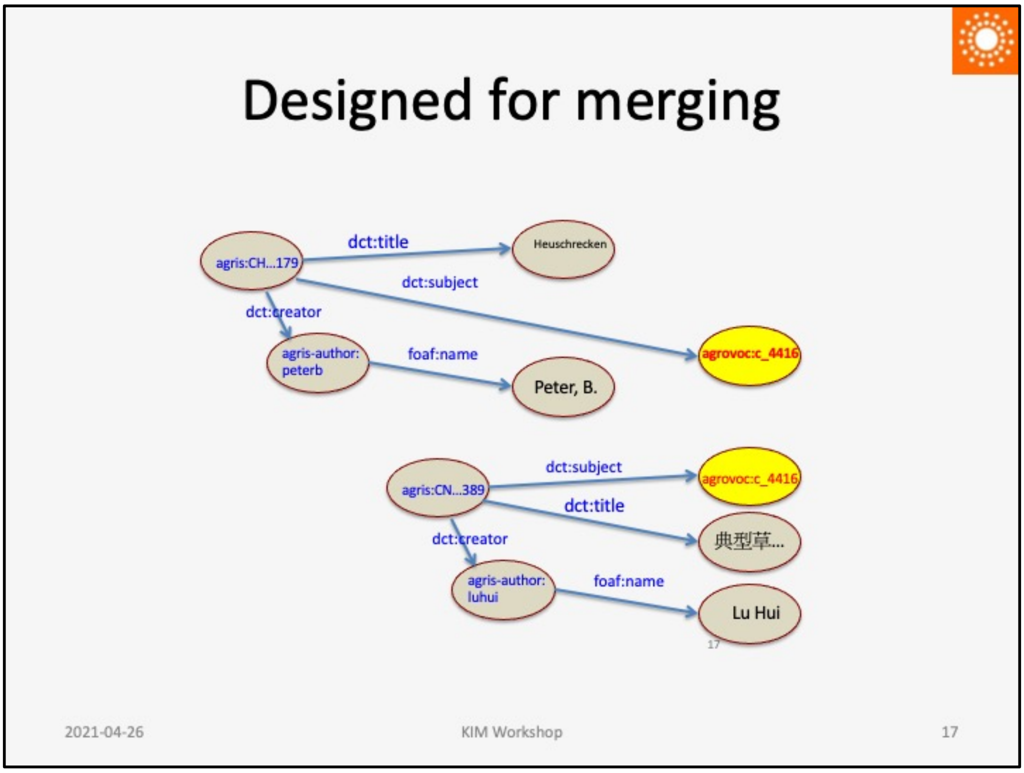
Encoding Schemes, first published in 2000, are still part of DCMI Metadata Terms, but they have been largely superseded by the development of SKOS concept schemes and by XML Schema datatypes, so were not included in the ISO standard.



For an extension of the ISO standard for Dublin Core, ISO 15836 Part 2, DCMI tweaked and clarified some aspects of the vocabulary.

The ISO standard is explicitly based on RDF but acknowledges its usability in non-RDF formats.

The extended standard introduces flexibility by softening some of the formal declarations of range classes to the less formal and more inclusive notion of "range includes". This provides non-binding guidance on how a given property is intended to be used.



The great strength of RDF is the ease with which data from different sources can be linked or merged.

Let’s take two simple graphs – in this case, two journal articles about grasshoppers, as represented by a URI from AGROVOC for the concept “grasshoppers”.

Designed for merging

Subject	Predicate	Object
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	dct:creator	agris-author:luhui
agris-author:luhui	foaf:name	Lu Hui
agris:CN2009002389	dct:subject	agrovoc:c_4416

18

We can take the triples for the first graph...

Designed for merging

Subject	Predicate	Object
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	dct:creator	agris-author:luhui
agris-author:luhui	foaf:name	Lu Hui
agris:CN2009002389	dct:subject	agrovoc:c_4416
agris:CH2001000179	dct:title	Heuschrecken brauchen ökologische Ausgleichsflächen
agris:CH2001000179	dct:creator	agris-author:peterb
agris-author:luhui	foaf:name	Peter, B.
agris:CH2001000179	dct:subject	agrovoc:c_4416

Add the griples for the second graph...

Designed for merging

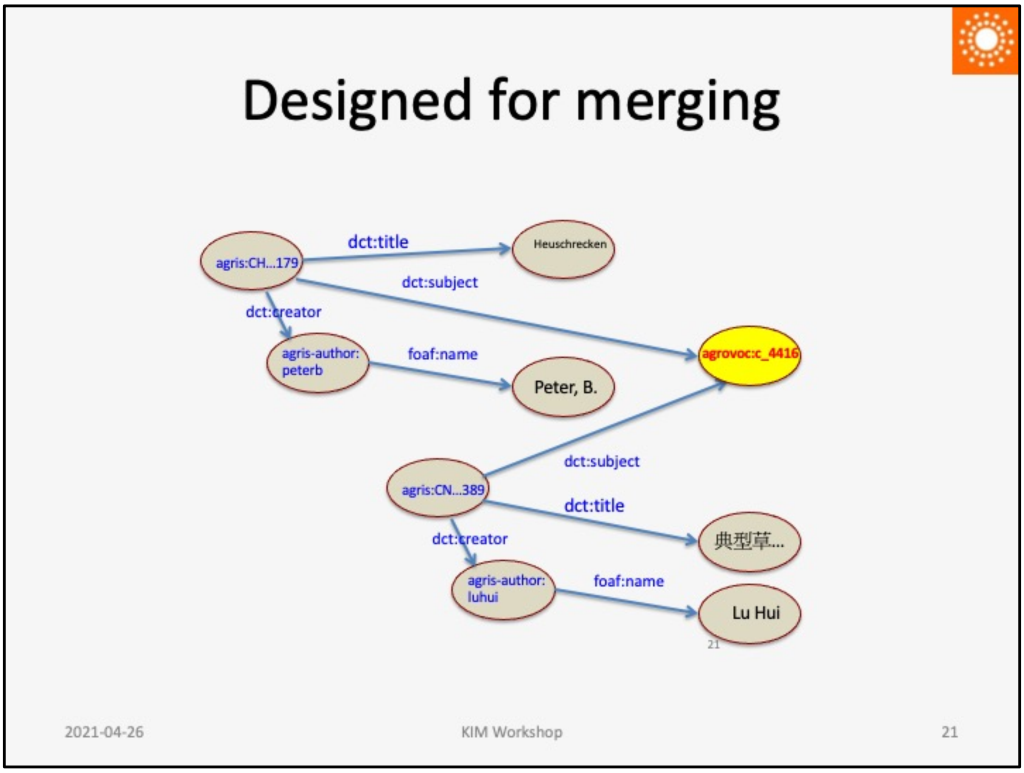
Subject	Predicate	Object
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	dct:creator	agris-author:luhui
agris-author:luhui	foaf:name	Lu Hui
agris:CN2009002389	dct:subject	agrovoc:c_4416
agris:CH2001000179	dct:title	Heuschrecken brauchen ökologische Ausgleichsflächen
agris:CH2001000179	dct:creator	agris-author:peterb
agris-author:luhui	foaf:name	Peter, B.
agris:CH2001000179	dct:subject	agrovoc:c_4416

20

2021-04-26 KIM Workshop 20

The computer can easily detect matching URIs.

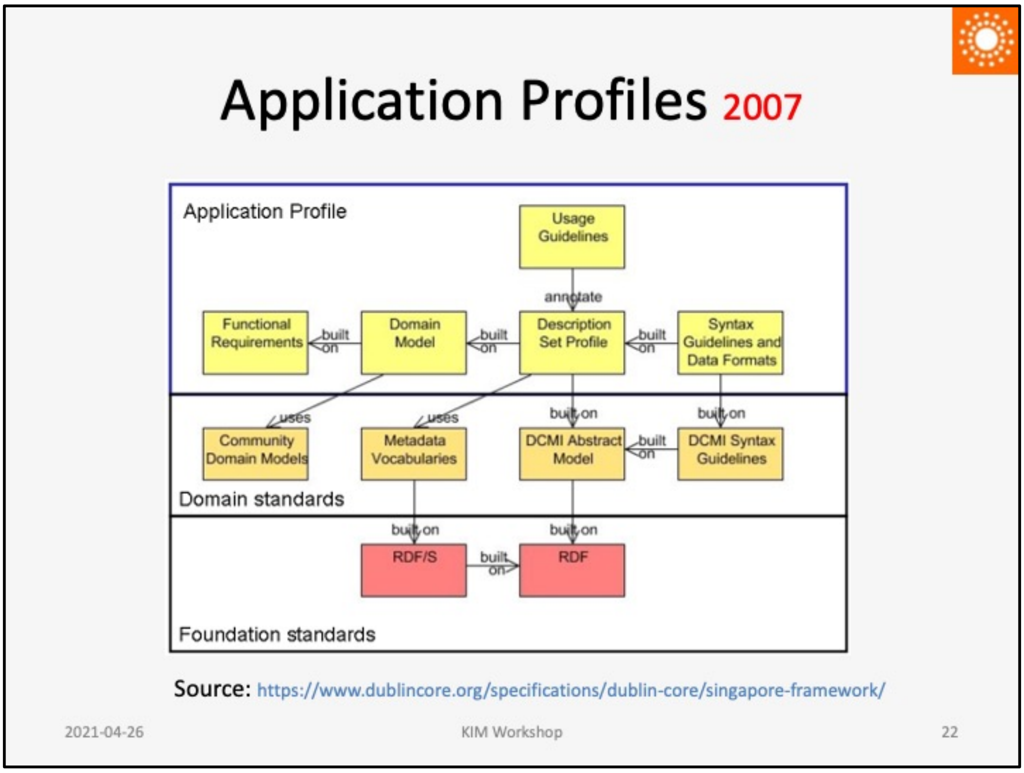
Computers can make these connections even if there are millions of triples.



This connection merges the two graphs into one bigger graph.

For merging data, RDF is awesome.

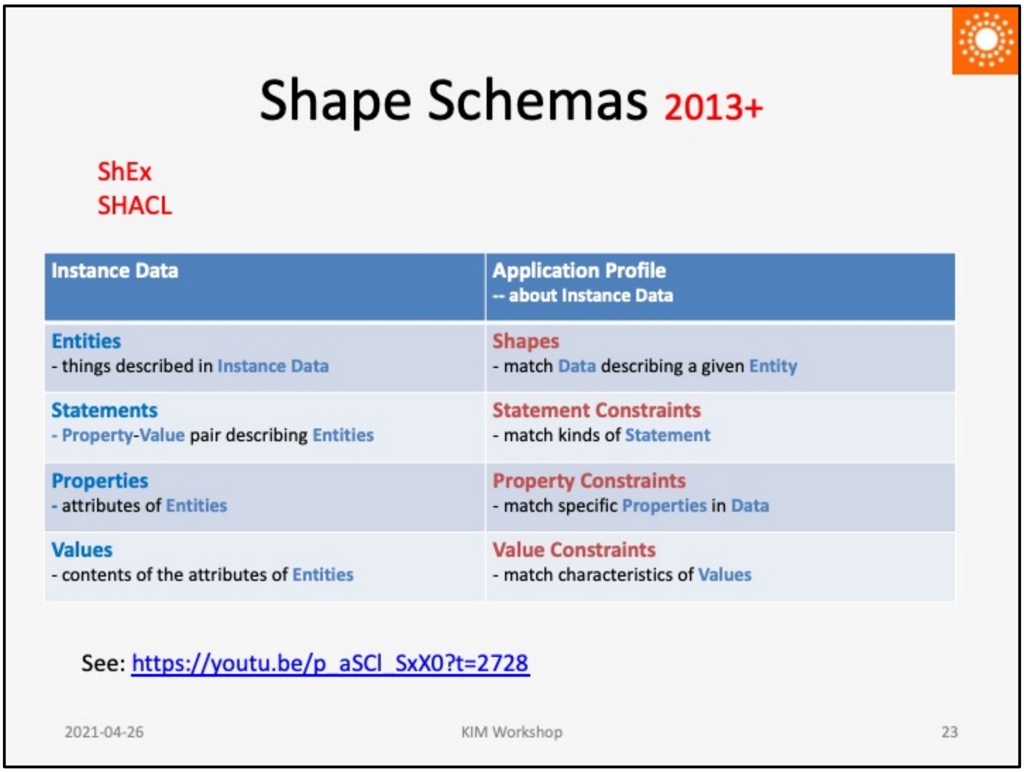
Until quite recently, however, developers and implementors had no easy way to validate Graph Data for consistency or for conformance with rules.



The DCMI notion of an Application Profile, first proposed in 1999, became the means by which the Dublin Core community sought to enable the validation of graph data.

In the Singapore Framework of 2007, Application Profiles use properties from one or more RDF vocabularies and constrain their use for a specific application.

This idea had a significant impact on metadata discourse but unfortunately did not lead to the development of practical software solutions.



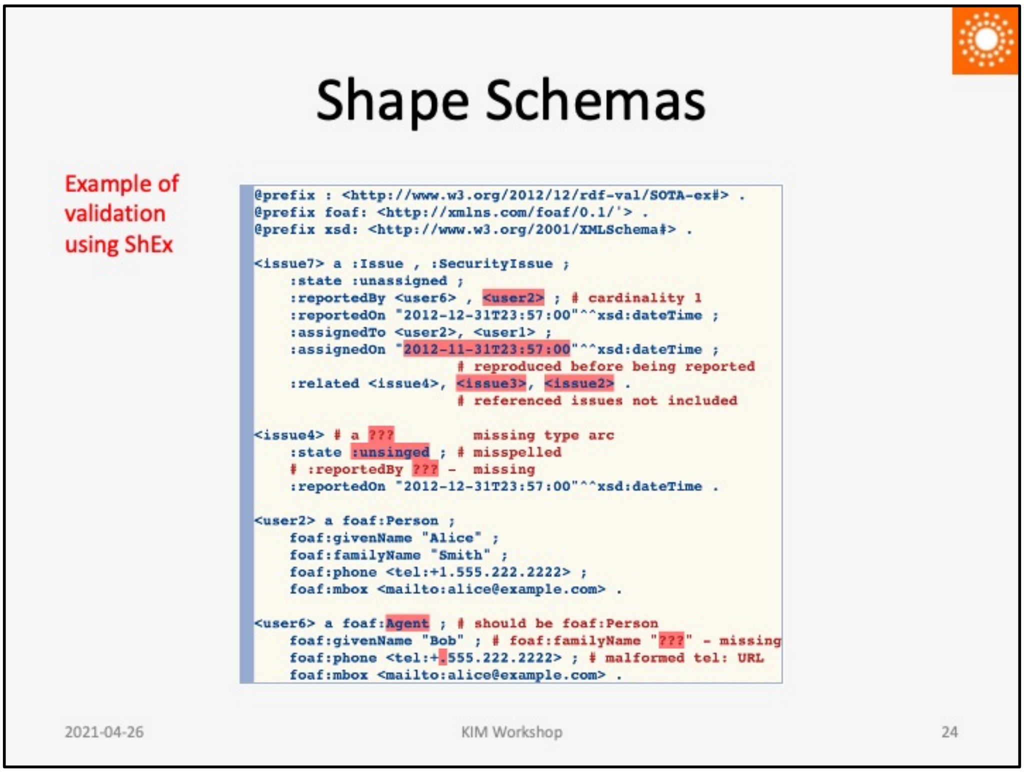
For that, we needed to wait several years for the development of RDF validation languages, notably SHACL and ShEx.

These languages are based on the concept of a Shape.

A Shape describes the properties, values, and constraints that one expects to find in instance data.

A shape can be matched to instance data in order to test whether that data meets expectations.

Application Profiles can be mpw expressed as Shape Schemas.



A ShEx schema is to RDF data what an XML schema is to XML data.

In this example, errors have been discovered in some data by matching it against the shape schema.

Shape Schemas in Spreadsheets

DC Tabular Application Profiles

shapeID	propertyID	mandatory	repeatable	valueNodeType	valueDataType	valueShape
:book	dct:creator					:person
	dct:date	Y	N		xsd:year	
	dct:title					
:person	foaf:name			literal		

See: https://www.dublincore.org/groups/application_profiles_ig/dctap_primer/

2021-04-26 KIM Workshop 25

Shape languages are very powerful, but learning them is like learning to drive. It takes more than a day or two.

In order to help people get past this obstacle, DCMI is developing a specification, DC Tabular Application Profiles, for creating a shape-compatible application profile in a spreadsheet.

The tabular format is designed to be convertible into shape schemas, and possibly into other types of schema.

If any of you would like to help with this, please get in touch.

Takeaways

- Pre-1995, metadata always used structures and formats specific to local contexts.
- The Web made it easy to access data globally, so people wanted to mash it up in new ways.
- Graph Metadata made mashups easier.
- Application Profiles and Shape Schemas bridge the gap between Format and Graph.
- RDF can serve as a Lingua Franca for both Graphs and Formats.

2021-04-26 KIM Workshop 26

Before the Web, metadata was Format Metadata. It was local to a specific context and there was no requirement for its structures and formats to work outside of that context.

When the Web arrived, it became possible to access data from many sources, and people wanted to mash it up in new and unexpected ways.

This need drove the development of RDF, a generic model based on statements that can be aggregated into graphs.

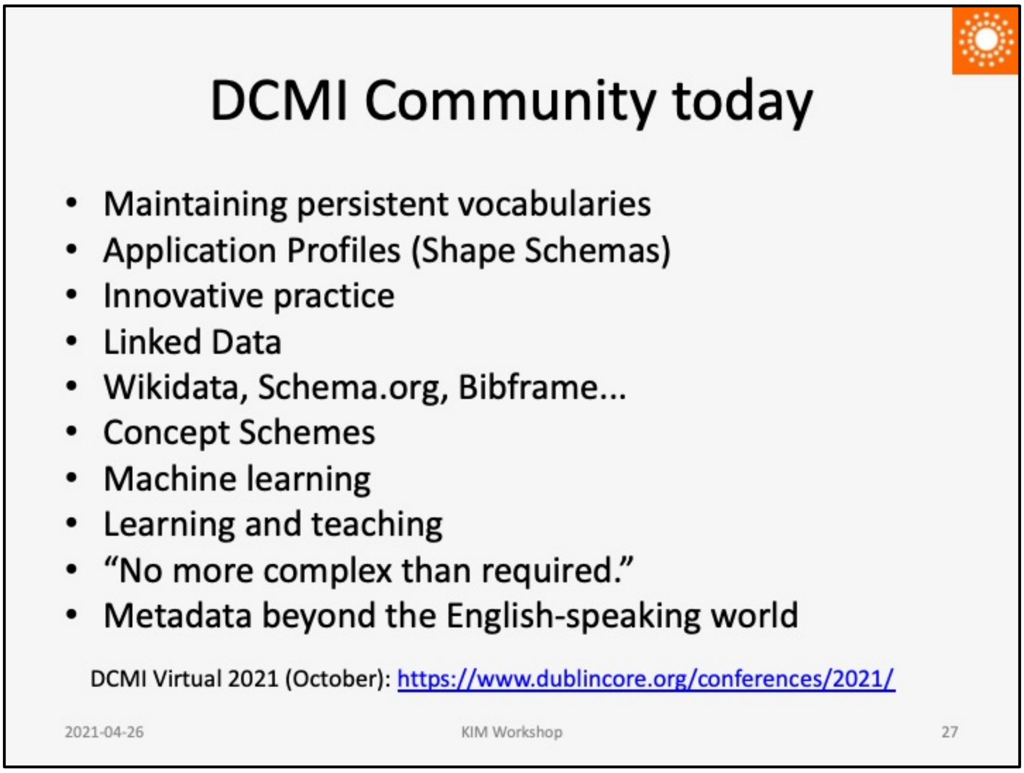
The gap between Format Metadata and Graph Metadata is being bridged by technologies such as Dublin Core application profiles and shape schemas.

Shape schemas make it possible to validate a graph as itif were a format.

Format Metadata and Graph Metadata can and will continue to exist side-by-side. Metadata technology will continue to evolve. All of the software we use today will eventually become obsolete.

In the meantime, RDF -- that is, URI vocabularies used with generic three-part statements -- can serve as a lingua franca with which formats and graphs can roughly interoperate, if only partially and imperfectly.

Our best hope for ensuring that today's metadata can be understood twenty years from now lies with the preservation of its underlying vocabularies.



The DCMI community today is about more than Dublin Core.

Over the course of twenty years, the focus has shifted to Application Profiles, to their interoperability on the basis of RDF and, more recently, to their expression as Shape Schemas.

It is about a forward-looking, experimental approach to metadata that one might summarize as “innovative practice”.