TDWG Technical Roadmap 2008

Technical Architecture Group | Convenor: Roger Hyam | 15th October 2008

What is a system? A system is a network of interdependent components that work together to try to accomplish the aim of the system. A system must have an aim. Without an aim, there is no system. The aim of the system must be clear to everyone in the system. The aim must include plans for the future. The aim is a value judgement.

William Edwards Deming

Introduction

The Technical Architecture Group of the Biodiversity Information Standards (TDWG) is charged with maintaining an overview of TDWG standards, their relationships to each other and to standards produced by other organisations. The charter of the TAG states that a yearly report will be produced called the 'Roadmap'. This is the 2008 Roadmap document. It supersedes the 2007 Roadmap presented in Bratislava and the 2006 Roadmap presented at the St Louis. Previous roadmaps are available on the TDWG website.

The reader is referred to the 2007 Roadmap for a fuller justification for the architecture and summary of all historical TDWG standards.

Standards Architecture

This section is produced verbatim from last years roadmap as an introductory overview to the architecture.

Why have a standards architecture?

From the point of view of exchanging data – such as in the federation of a number of natural history collections – there is no need for a standards architecture. The federation is a closed system where a single exchange format can be agreed on. The federation can grow by adding new members whose needs are met by the format. This model has worked well in the past but it does not meet the primary use case that is emerging. Biodiversity research is typically carried out by combining data of different kinds from multiple sources. The providers of data do not know who will use their data or how it will be combined with data from other sources. The consumer needs some level of commonality across all the data received so that it can be combined for analysis without the need to write computer software for every new combination. This commonality needs to seamlessly extend to new types of data as they are made available. An architecture is required to provided this commonality.

What form should the architecture take?

A degree of commonality could be achieved simply by specifying how data should be serialised. If all suppliers passed data as well-formed XML, for example, it would provide a degree of interoperability but clients would still not know how the elements within one XML document relate to those in another or how the items described in those documents were related. At the other extreme, the architecture could provide a detailed data type library which described the way in which each kind of data should be serialised at a fine level of granularity – which XML elements must be present and what they should contain – but it is highly unlikely that a single set of serialisations would meet all needs any more than a single federation schema would. It remains a requirement of some thematic networks that they have well defined data types to ensure that the data passed is valid and fit for purpose.

The architecture therefore has to meet two needs. It has to allow generic interoperability but also restricted validation of data for some networks. It does this by taking a three pronged approach.

1. An ontology is used to express the shared semantics of the data but not to define the validity of that data. Concepts within the ontology are represented as URIs (Universal Resource Identifiers).

2. Exchange protocols use formats defined in XML Schema (or other technologies) that exploit the URIs from the ontology concepts.

3. Objects about which data is exchanged are identified using Globally Unique Identifiers.

This means that (although exchanges between data producers and clients may make use of different XML formats) the items the data is about and the meaning of the data elements is common across all formats.

Globally Unique Identifiers (GUIDs)

GUIDs are a key component of the architecture because they solve two main problems:

● **Multiple Routes** In an environment where information is shared freely a client (either machine or person) may receive the same piece of data via different routes and believe it has received multiple pieces of data. One piece of data may say that species A occurs at longitude X and latitude Y and another may say A occurs at long. X lat. Y with a

error of 1km. Even if both these pieces of data have been retrieved from the same data source there is no way of knowing if they are referring to the same or separate measurements of the real world. i.e. Do they add to our confidence that A occurs at X,Y or not? Only if the two pieces of data are given identities by their originators (and linked back to their sources should they be derived) can we track whether these are the same piece of data or derived from a common piece of data or even from each other. The requirement for GUIDs is clear even without defining what constitutes a valid "piece of data". Without GUIDs it is even hard to know how much biodiversity data is in circulation, and claims should be treated with caution.

- **Ownership and Trust (Provenance)** This is closely related to the multiple routes problem. When a client (either machine or person) receives a piece of data how do they know whether it has been 'corrupted' or 'improved' by a third party? Whether it is up to date or whether it has been updated or deprecated? How do they know what they can legally do with it in terms of disseminating it and, most importantly, how do they credit the producers of the data? The simplest and only workable solution to this problem is for the piece of data to carry with it a means of retrieving a clean copy of itself and any associated legal metadata. Tagging the data with a resolvable (i.e., can be dereferenced) GUID allows this to happen. The most familiar example of such a GUID is the URL of a web page. A client may receive a printed or cached web page but can always get back to the original version by calling its URL.

These points and others were discussed during a series of international workshops and meetings over the last three years that resulted in a proposal to adoption Life Science Identifiers (LSIDs) as the preferred technology for GUIDs. The details of the use of LSIDs are outlined in the LSID applicability statement that is currently proposed as a TDWG standard.

The TDWG architecture does not mandate the use of LSIDs. The only requirements of GUIDs is that they really are globally unique and that they are resolvable, ideally through standard web technologies such as HTTP. The LSID applicability statements recommends the use of an HTTP proxy for LSIDs which makes them behave much like regular URLs. Similar proxy approaches are taken with other GUID technologies such as DOI and older standards such as ISBN or non-resolvable GUIDs, such as Universally Unique Identifier (UUIDs).

GUID Challenges

Adoption of GUIDs is the single most important requirement to improve the quality of biodiversity data

exchange and yet data suppliers do not appear to be adopting them with any sense of urgency. Primary reasons for this are:

- Lack of knowledge, understandably caused by seeing the problem from a data suppliers point of view rather than as the consumer of data from multiple sources.

- Lack of permanent internal identifiers for data. e.g. they may rely on database primary keys that are changed during database migrations but which don't affect their internal working processes - a spreadsheet mentality to data.

- Lack of the awareness of the primary benefit to the data supplier of tagging data with GUIDs e.g. the ability of users to cite their sources and give credit even when data from multiple sources has been used in an analysis.

- Confusion over GUID standards. There are a series of competing standards and there has been much open debate as to which to adopt. This even includes confusion within standards on how they should be implemented e.g. data return types.

- Difficulty in implementing LSIDs. LSIDs require the addition of a special DNS rule to the name server that controls the institution's domain name. Data curators may not have access to this name server or even know what a name server is, let alone how to change it. This is a major barrier to LSIDs adoption.

- Difficulty in implementing any resolvable identifier - even the URL to a web page. Many institutions have corporately managed web sites and it is no simple matter for data curators to add database-driven content, even if they have the skills and applications/tools to hand.

It appears the discussions on GUIDs within TDWG have not addressed the true level of resources available in the community.

TAG GUID Recommendations to Data Suppliers

- You must have working practices that maintain locally unique IDs on your data within your databases. If you don't do this you will not be able to expose your data using globally unique IDs now or in the future.

- Whenever you expose your data you should include GUIDs that can be resolved back to your internal, locally unique IDs. Don't get hung up on the technology. If you can't manage LSIDs, some type of URL that your institution can commit to maintaining for the foreseeable future is fine. In simple terms - every 'thing' you expose to the world should be represented by a web 'page'.

Ontology

"Ontology" is a loaded term with many definitions connected to creating formalisations of reality - a subject that has occupied thinking people for thousands of year. The TDWG ontology is more of a functional thing and, wearing retrospectacles, the word 'Dictionary' would have been a better one to use although this also has multiple meanings.

If GUIDs are used to uniquely identify 'pieces' of data

we need to have a shared understanding of what we mean by a 'piece of data' i.e. what kind thing is it that a particular id applies to, a specimen, a person, an observation, a complete data set. We also need to have a shared understanding of at least some of the properties we use to describe these things. This is the function of the TDWG ontology. It is not an expansive formalisation of the biodiversity informatics domain but

a rather trivial list of the things that we, as a community, can agree on the meaning of.

The TDWG Ontology enables the following kind of interaction. A client receives a piece of data such as "Species A occurs at longitude X, latitude Y" that is tagged with a GUID. The client application resolves the GUID back to the originator of the data (just like looking up a web page) and receives the data associated with the GUID. This includes the fact that the GUID is for a type of thing that is a tdwg:Specimen that is located in a particular tdwg:Collection. On the basis of this client can take appropriate action that may be different from the actions it would have taken if it the GUID had been for a object of the kind tdwg:Observation.

If there are multiple exchange standards (catering to multiple application requirements) that all map to the TDWG Ontology (and other well known vocabularies) using XML namespaces then it is possible for application developers to "cross-walk" between different standards and build useful tools. If exchange standards do not map to the TDWG Ontology then everybody who needs to translate between exchange standards is likely to do it slightly differently and precision will be lost.

Ontology Challenges

The word ontology is a major challenge as it has dragged the community into hours of discussions concerning RDF, OWL, reification and inference. In so doing we lose sight of the tremendous benefits of having a basic list of shared objects and their properties.

Even to keep a list of core TDWG concepts up to date, manage the consensus building process around new concepts and educate standards developers in how to integrate the ontology into their proposals is a very time consuming and therefore expensive process. Nobody has been resourced to do this work in 2008 and therefore it hasn't happened as it should. Unless some form of ontology manager is given the resources to curatate this central resource then there is a danger that the expectations of interoperability will not be met.

TAG Ontology Recommendations

- Don't get hung up on complex talk of ontologies and inference especially if you are not designing your own exchange formats. The ontology should be thought of as a concept repository for the community not a model of the entire subject domain.

- If you are working on an exchange format don't make up new concepts if you can help it. Use existing ones from the TDWG ontology, IETF, W3C, Dublin Core etc or at least describe how your concepts match to these.

- If you are using XML use namespaces correctly so as to exploit other vocabularies. If you can't do this, publish a mapping to well know namespaces.

Exchange Protocols

TAPIR

The most well developed area of TDWG before the standards architecture was proposed were the exchange protocols DiGIR and BioCASe with their associated federation schemas DarwinCore and ABCD. This momentum has continued with the development of the TAPIR protocol that unifies the DiGIR and BioCASe protocols and also comes closer to the OGC's WFS protocol. Because TAPIR, in its more complete implementations, allows the specification of output data models it is possible for TAPIR based providers to mimic other types of data provider.

There were two important developments related to TAPIR in the last year. One of them is a new service (TapirTester) to test if a TAPIR provider is compliant with the current TAPIR specification. It can be used by those who need to implement new data provider software or by users of data provider software who want to check if their services are working properly. Although TapirTester does not include all possible tests that can be performed, it covers most aspects of the protocol, which makes it an important quality control tool. The new service is available both as a web interface and as a web service.

The other development (TapirBuilder) facilitates the creation of TAPIR documents. TAPIR networks depend on specific documents to work: XML Schemas (that can be created by existing tools), output models and query templates. The last two documents are specific to TAPIR and sometimes it can be difficult to produce them by hand. TapirBuilder is a new online tool to help building such documents.

A new format based on XML was also proposed to represent data abstraction layers for TAPIR (replacing the previous CNS configuration files using key-value pairs in plain text). An index containing some of the existing data abstraction layers is available. There is also discussion on the need to be able to "discover" existing TAPIR models that people develop as these would be useful for other people to reuse, when appropriate.

Current plans are to submit TAPIR to the TDWG standards track by the end of 2008 after final discussions in the TAPIR mailing list.

Delimited Files

While harvesting completed sets of flat DarwinCore type data for the GBIF data portal it became apparent that the communications using existing TDWG protocols (DiGIR, ABCD and TAPIR) were very verbose, resulting in a large amount of network traffic and database activity to do something that could be achieved in a simpler manner. Additionally, there are data providers who have either

- too large a dataset to effectively harvest the full set in one go over existing protocols (months of work)

- or do not have the ability to install wrapper software on an accessible web server.

For these reasons, it has been decided to support simple delimited files within the GBIF harvesting mechanism, that represent the full dataset, and are produced on the provider side using a simple database export and then compressed for transfer. It is proposed that extensions to simple occurrence records will be

supported on a one file per extension basis, whereby the row in the extension file references an identifier in the core file, thereby allowing for "many to one" style extensions. This is effectively creating a relational model.

Importantly an attempt will be made to map the files and the fields within the files to the TDWG ontology and to provide GUIDs for the rows in the core file. In this way these delimited files will leverage the TDWG architecture for both very large and very small data sets.

This is likely to be an area of active development in 2009.

REST Services

There has been some discussion on REST (Representational State Transfer) web services. REST is a style of architecture rather than a specific protocol. Both TAPIR and OAI-PMH could be regarded as RESTful services. Where the REST pattern is not matched by the TDWG architecture is the degree to which GUIDs are not all uniquely addressable using a universal syntax. Both LSIDs and DOIs make use of independent resolution mechanisms. It is therefore recommended, and is common practice, to provide a version of these GUIDs that use HTTP proxy resolution. **HTTP should be thought of as the universal syntax for addressable resources.**

TAG Protocol Recommendations

- If you want to expose data to a particular project follow that particular project's current recommendations. It is the role of the TAG to make recommendations to projects not to individual data suppliers.

- If you are making the recommendations for a project and have a need for queryable nodes within your network then recommend TAPIR rather than DiGIR or BioCASe.

- Even if you exchange plain XML documents using TAPIR you should make sure you use GUIDs and map to the TDWG Ontology where possible.

- If you are making the recommendations for a project and only require a harvesting protocol then consider using OAI-PMH but discuss with the TAG what your metadata formats may be so as to enhance re-use. No one is harvesting with OAI-PMH in our community at the moment.

- If you are making the recommendations and it is likely that some or all of your data suppliers will not be able to set up and run web services consider participating with Tim Robertson and Markus Doring at GBIF in development of a delimited file standard.

Resources

Roadmap 2007	http://wiki.tdwg.org/twiki/pub/TAG/RoadMap2007/TAG_Roadmap_2007_final.pdf
TDWG Main Website	http://www.tdwg.org/
TAG Wiki	http://wiki.tdwg.org/TAG/
TAG Home Page	http://www.tdwg.org/activities/tag/
TAG Mailing list	http://lists.tdwg.org/mailman/listinfo/tdwg-tag
GUID Task Group	http://www.tdwg.org/activities/guid/
TDWG Ontology (starting at TaxonName)	http://rs.tdwg.org/ontology/voc/TaxonName
TAPIR Task Group	http://www.tdwg.org/activities/tapir/
Dublin Core Metadata Initiative	http://dublincore.org/
Darwin Core Wiki	http://wiki.tdwg.org/twiki/bin/view/DarwinCore/WebHome
OAI-PMH	http://www.openarchives.org/OAI/openarchivesprotocol.html
TAPIR	http://www.tdwg.org/activities/tapir/
TapirBuilder	http://tapir.tdwg.org/builder
TapirTester	http://tapir.tdwg.org/tester
DiGIR	http://digir.sourceforge.net/
BioCASe	http://www.biocase.org/