International Union for
Biological Sciences
Taxonomic Databases Working Group
http://www.tdwg.org

TDWG Technical Architecture Group
11th to 13th April 2006
eSI Edinburgh

1.Summary

Fourteen TDWG members attended the meeting hosted by Jessie Kennedy. The goals were to review the current technologies, initiate a discussion on a TDWG standards architecture and make recommendations on moving towards the such an architecture.

The review of current technologies is included as Appendix B of this report.

Sharing data across a network implies: tracking ownership, reporting usage, remote referencing, repeatability and detecting duplication etc. These features require that units of data must be identifiable. The architecture envisaged therefore consists of shared biodiversity data being modelled as a graph of identifiable objects. The semantics of these objects (i.e. their types) will be encoded in shared ontologies that are linked by central 'core' and 'base' ontologies.

A major risk was identified in that the current infrastructure supported by DiGIR/DarwinCore and BioCASe/ABCD has not been established with the intention of publishing identifiable objects. Many data providers do not have stable internal identifiers for records and have not considered GUIDs (globally unique identifiers). It will take time for these providors to accommodate changes implied in the use of GUIDs.

Jessie Kennedy's team will investigate seeding an ontology based on the existing schemas by the end of the summer 2006. This ontology will impact TAPIR and DiGIR2 developments as they could make effective use of the ontology to enable automated comparison of data retrieved from the two types of provider. Existing deployments should continue but thought should be given to support of GUIDs within the current infrastructure.

Attendees are listed in Appendix A. Further information about the meeting, including PowerPoint presentations can be found on the meetings wiki pages[1] or contact Roger Hyam (roger@tdwg.org)

2.Goals

a. Review Current TDWG and associate technologies.

b. Discuss a high level TDWG standards architecture

   - The goal of a TDWG Architecture is to provide a framework which TDWG and external standards can interrelate .

[1] http://wiki.tdwg.org/twiki/bin/view/TAG/TagMeeting1

  - A TDWG standards architecture assumes that the goal of TDWG in this regard is to facilitate information transferr between agents (individuals, projects and organisations).
  - A TDWG standards architecture also assumes transfer in time (archival and retrieval)..

c. Make recommendations to facilitate a transition to to a TDWG standards architecture.

3.Outcome

A review of the current technologies is given in Appendix B. Agreement was made on the following points during the course of the meeting.

3.1.High Level Vision of a TDWG Standards Architecture

a. The architecture is concerned with shared data.

b. Biodiversity data will be modelled as a graph of identifiable objects.

c. The semantics of these objects will be encoded in a series of shared ontologies.

d. Ontologies will be related to each other on the basis of a shared Base and Core ontologies as a minimum.

e. A series of interfaces/protocols will specify how services on the network will expose objects.

f. These interfaces/protocols will preferably be adopted from existing technologies but created by TDWG as necessary.

g. Standards will define how objects should be serialized for exchange over the network.

h. Sharing data across a network implies: tracking ownership; reporting usage; remote referencing; repeatability; deduplication etc. While not a panacea, GUID technologies support these requirements.

i. **Recommendation:** The GUIDs Group should issue a document clearly justifying adoption of GUID technology. The advantages need to be clearly explained to the wider community.

j. **Risk:** The current infrastructure supported by DiGIR/DarwinCore and BioCASe/ABCD has not been established with the intention of publishing identifiable objects. Many data providers do not have stable internal identifiers for records and have not considered GUIDs. It will take time for these providers to adapt.

k. **Recommendation:** Data providers established from now on, even using existing technologies, must consider how they will support GUIDs in future.

3.2.Data Not in 'Identifiable Objects'

a. Not all data will be shared in identifiable objects.

b. Results of a query (DiGIR , SQL, SPARQL, TAPIR for example) are not identifiable objects.

3.3.Objects

a. Objects are identifiable and machine readable.

b. Objects should be semantically rich but can have opaque binary components.

c. The result of resolving a GUID will be an object.

d. There needs to be a simple way of identifying the type of these objects.

3.4.Typing Objects

a. **XSD:** In XML Schema specification, the supply of a schema location within a instance document is not required. In TDWG All XML objects defined by XSD must have a schema location that resolves to a standard schema that the object will validate against. TDWG must permanently host schemas that are considered standards but application providers may host their own schemas if they need to.

b. **RDF:** All TDWG objects in RDF/S must have at least one rdfs:type property.

c. Libraries need to handle RDF embedded in XHTML.

d. **Recommendation:** RDDL (http://www.rddl.org/) needs to be assessed for possible adoption as a standard technology for organising object definitions.

3.5.Definition of Objects 1 - XSD

a. The object structure must be defined as a top level element (current schemas would have to be modified).
b. Top level elements that define objects should be defined by global complexTypes - this allows automated tools to build binding code.
c. Whatever these top level objects are they must have a GUID attribute.

d. **Recommendation:** A standard pointer structure must be defined and adopted to reference objects defined in XSD.

3.6.Definition of Objects 2 - Semantic Web Technologies.

a. An object is an instance of a class in an ontology.
b. Objects should be bounded by Concise Bounded Descriptions and identified by a GUID.
c. Anyone can make assertions about a resource but the definitive form is the one that is returned when the GUID is resolved.

d. **Recommendation:** The minimum properties of an object need to be defined - perhaps as part of a base class. These will include a human readable string and an rdfs:type property.

3.7.Data Modelling

a. This is key to integration of TDWG standards.

b. UML accompanied by natural language descriptions should be used to discuss and define the TDWG object model.

c. Conceptually there will be three levels in the ontology:

- Base = Abstract base class and properties for all TDWG objects. (e.g. GUID, title etc)
- Core = Extends base to define classes and properties that are common to multiple domains.
- Domain Ontologies[2] = Concrete classes for use.

d. One of TAG's roles is to ensure redundancy does not creep into new standards/ontologies.

e. Classes and properties within the Base and Core ontologies will have a status attribute that indicates their level of stability/adoption.

f. **Recommendation:** All subgroups should consider presenting data modelling as natural language descriptions accompanied by UML diagrams.

g. **Recommendation:** All data models should extend the Base and Core ontologies and make use of existing ontologies.

h. **Risk:** Exchange of UML diagrams other than as pictures may be problematic because of interoperability issues between UML tools.

i. **Risk:** It may be desirable to use modelling constructs that are not supported by UML.

j. **Action:** Jessie Kennedy's group to coordinate development on non-normative 'first-pass' ontology from existing schemas and make recommendation for proceeding with base and core ontologies within the next three months.

k. **Recommendation:** Multiplicity relationships may be key to identifying primary objects.

l. **Action:** Jessie Kennedy's group to examine conversion of UML to semantic web and XML Schema representations.

m. **Risk:** It is acknowledged that managing ontologies through time may prove complex and costly.

3.8.GUIDs

a. There is a clear line between classes and instances (ontology and data) but this line will be in different places depending on the application. Some people may consider taxon concepts as classes **or** descriptive terms etc...

b. There are certain things for which LSIDs (an example of a GUID) are not appropriate. It would be legal to use LSIDs for RDF resource identifiers for class and property definitions and XML Schema locations but existing software libraries would have to be extended, and this is not desirable.

c. **Recommendation:** LSIDs should not be used for ontologies or XML Schema locations. LSIDs should be used to refer to instances. [This recommendation has subsequently been debated on the TAG mailing list. It should, perhaps, be a matter for the GUID group to resolve].

d. **Recommendation:** LSIDs should be limited to URI not IRIs at the moment.

3.9.Services provided by a network node.

a. The minimum requirements for a data provider node were discussed in terms of four levels: Resolution, Harvest, Search and Query.

b. In the future Resolution should be the minimum service supplied by data providers for example LSID resolution. (See under point 3.1)

c. It would be very useful for providers to implement some form of harvesting and/or syndication service as this would enable other entities to layer thematic services on top of limited providers.

d. **Recommendation:** There is an urgent need to adopt an existing harvesting protocol that could be used alongside GUID resolution. OAI and other technologies should be considered.

e. It was not possible to clearly differentiate between Search and Query; different search and query protocols will co-exist. See Appendix B.

3.10. Namespaces and Resource Locations

a. All technologies under consideration require that some files are hosted in a permanent, project-neutral location.

b. **Recommendation:** TDWG should provide a permanent project-neutral location for its existing and emergent standards. The location should be at http://res.tdwg.org/ (res standards for resource).

c. Working groups should be given their own space within this permanent repository to host files such as XML Schema locations, RDF and OWL ontologies.

d. **Recommendation:** Namespaces currently used in TDWG standards should be surveyed and a policy of changing them and/or adding redirects within the infrastructure established.

3.11. Near Term Advice to Implementers

a. **Recommendation:** We do not recommend short term replacement of existing technologies as their potential replacements are not mature. This does not include the already scheduled roll out of TAPIR and DiGIR2.

b. **Recommendation:** Any new deployments or changes to deployments should address the need for migration to GUID based technologies in the near future.

4. Conclusions

The meeting was considered a success by participants. It was possible to produce a clear list of points on which agreement could be reached, to identify areas where agreement could not be reached and to make clear recommendations. The formation of a Technical Architecture Group for TDWG is required. The TAG is needed to provide technical guidance and advice.

Appendix A: Attendees

Name	Email	Institution
Dave Vieglais	vieglais@ku.edu	Biodiversity Research Center Informatics, University of Kansas, USA
Greg Whitbread	ghw@anbg.gov.au	Australian National Botanic Gardens
Gregor Hagedorn	G.Hagedorn@bba.de	Institute for Plant Virology, Microbiology, and Biosafety Federal Research Center for Agriculture and Forestry (BBA), Germany
Herbert Schentz	herbert.schentz@umweltbundesamt.at	Umweltbundesamt, Austria
Javier de la Torre	jatorre@gmail.com	CSIC, Madrid, Spain
Jessie Kennedy	J.Kennedy@napier.ac.uk	e-Science Institute, Edinburgh, UK
John Wieczorek	tuco@berkeley.edu	UC Berkeley, USA
Katharina Schleidt	katharina.schleidt@umweltbundesamt.at	Umweltbundesamt, Austria
Markus Doring	m.doering@BGBM.org	The Botanic Garden and Botanical Museum Berlin-Dahlem, Germany
Renato Giovanni	renato@cria.org.br	Centro de Referência em Informação Ambiental, Brazil
Rob Gales	rgales@ku.edu	Biodiversity Research Center Informatics, University of Kansas, USA
Robert Kukla	R.Kukla@napier.ac.uk	Napier University, Edinburgh, UK
Roger Hyam	roger@tdwg.org	TDWG, RBGE Edinburgh, UK
Steve Perry	smperry@ku.edu	Biodiversity Research Center Informatics, University of Kansas, USA

Appendix B: Summary of Current Technologies

1.Current Technologies and Their Relationships.

TDWG standards are not directly related to the protocols and encoding schemas that are currently used to exchange data in the biodiversity domain. This is a concern.

This section lists the major technologies currently in use to exchange data by TDWG members and those that are about to be deployed. More detailed information on DiGIR, BioCASE and TAPIR can be found in the initial TAPIR proposal from which some of the information below was taken[2].

Definitions followed here:

- A protocol is a mechanism for exchanging messages. The DiGIR, BioCASE and TAPIR protocols are more than just protocols in the sense of the HTTP protocol in that they also include a query syntax.
- A provider is an application for exposing data using a protocol.
- A data provider is an instance of a provider application.
- A Conceptual Schema is an XML Schema (XSD) that is used to define a data model for exchange of data.

1.1.Z39.50 – Protocol

"Z39.50" refers to the International Standard, ISO 23950: "Information Retrieval (Z39.50): Application Service Definition and Protocol Specification", and to ANSI/NISO Z39.50. The Library of Congress is the Maintenance Agency and Registration Authority for both standards, which are technically identical (though with minor editorial differences). This protocol was used for the Species Analyst Network (originally 120 data providers) but has now been replaced by the DiGIR Protocol.[3] It is the starting point for the development of DiGIR, BioCASe and TAPIR protocols.

1.2.DiGIR - Protocol

DiGIR[4] was conceived as a replacement for the Z39.50 protocol. The Z39.50 protocol was considered too complicated resulting in a steep learning curve for developers and including barriers to acceptance by network administrators.

DiGIR conceptual schemas are XML Schemas that define a flat list of non-nested elements that directly extend the DiGIR XML Schema via use of substitution groups. Arbitrary XML Schemas can not be used. DiGIR uses DarwinCore as a default conceptual schema though it could be used with other schemas provided they were specifically designed or adapted for the purpose.

There are widespread software implementations that use the protocol:

- **DiGIR PHP Provider –** A software application widely deployed by MaNIS, OBIS, GBIF and others, implementing DiGIR and used to serve Darwin Core.
- **DiGIR Portal -** A software application written in Java that federates searches across multiple DiGIR providers.

[2] http://wiki.tdwg.org/twiki/bin/viewfile/TAG/TAPIR?rev=1;filename=newprotocol.pdf

[3] http://www.loc.gov/z3950/agency/

[4] http://digir.sourceforge.net/

- **GBIF Data Repository Tool -** A package including Zope with a Python implementation of the DiGIR provider and a MySQL database with pre-configured tables.
- **Biota Provider -** A functional Java implementation of a DiGIR provider.
- **GBIF Portal Library –** Java libraries used by the GBIF data portal to communicate with providers.
- **Perl client Library –** Limited library used by species link for metadata and search messages only.
- **Globus DiGIR wrapper –** A web service developed by the SEEK project on top of the Globus toolkit to communicate with portal engines.
- **KE EMu –** A Commercial supplier of collection management software[5] is committed to releasing support.
- **Specify** – An open source collections management package[6] now supports a DiGIR interface.

There are currently 38 different institutions using DiGIR to serve 102 collections with something over 24 million records between MaNIS, ORNIS, HerpNet, and FishNet2 networks at the moment. There are 77 institutions with168 collections that are committed to participation and will be connected by the end of 2007.

DiGIR2 is the working name of an application developed at Kansas. It is not a new version of the protocol although it may serve data in DiGIR 1.x , TAPIR or other protocols in future. See below.

1.3.BioCASe – Protocol

The BioCASe protocol was developed and deployed as part of the BioCASE project[7] in 2003. The project had intended to use DiGIR but the primary conceptual schema for the project, ABCD, was too complex for the DiGIR protocol to handle. The BioCASe protocol binds to conceptual schemas by simple XPath expressions which loses strong XML based validation but allows hierarchical schemas.

There is a single implementation of the BioCASe protocol data provider in Python - the PyWrapper[8]. There are around 100 installations of provider packages using 4 different conceptual schemas chiefly in Europe. There are 4 client implementations:

- **Simple UI –** A prototype portal for federated searches. Used by BioCASe project and GBIF Germany.
- **Querytool –** A XSLT base query interface to a single BioCASe provided. Distributed as part of the PyWrapper package.
- **Synthesis Portal –** A multilingual XSLT based portal that will cache data. Currently in preparation.
- **GBIF data portal –** A large central indexer.

1.4.TAPIR – Protocol

[5] http://www.kesoftware.com/emu/index.html

[6] http://www.specifysoftware.org/Specify

[7] http://biocase.org/

[8] http://www.pywrapper.org/

The TAPIR[9] protocol has been developed as a unifying protocol to eliminate the differences between DiGIR and BioCASe. The first implementation of TAPIR is based on the PyWrapper used for BioCASe. TAPIR is near to version 1 and is currently going through a documentation phase that should be completed this summer. Several BioCASe providers are keen to adopt TAPIR including the Germplasm Clearing House mechanism[10]. TAPIR offers a number of powerful features that allow it to mimic other protocols such as KML[11]

1.5.DarwinCore – Conceptual Schema

DarwinCore[12] is a simple conceptual schema consisting of 49 elements though there are a number of variants that add extra elements to this set. It is the main schema used by the DiGIR protocol.

There are 3 main variants of the DarwinCore:

- 1.2 – Generic version.
- 1.21 - Used by FishNet2, HerpNet, ORNIS and MaNIS networks
- OBIS - Used by the Ocean Biogeographic Information System

There are other variants in use notably one by PaleoPortal. Version 2 is currently under development but progress is pending the outcome of the TAG. A mechanism is needed to unify the different flavours of DarwinCore that are available.

1.6.ABCD – Conceptual Schema

The Access to Biological Collections Data (ABCD) Schema is an evolving comprehensive standard for the access to and exchange of data about specimens and observations[13]. It was initially developed as part of the BioCASE project. It contains around 925 mapping-concepts (equivalent to elements) or nearly 1800 when EFG (Extended for Geosciences) is included. Unlike DarwinCore it defines a true hierarchal document structure. It is the principle conceptual schema used by the BioCASe protocol and will be served using the new TAPIR protocol.

1.7.BioCASe Provider

Software deployed by the BioCASE project, implementing BioCASe and used to serve data based on ABCD and other conceptual schemas. It can also serve Species2000 SPICE. It is the forerunner of PyWrapper v2.

1.8.PyWrapper v2 Provider

PyWrapper v2 will be the first implementation to support the TAPIR protocol It is designed to support multiple protocols. These will include TAPIR, WFS, Species 2000 SPICE, BioMOBY and maybe the OAI Harvesting protocol.

The PyWrapper works by mapping incoming requests under implemented protocols like TAPIR into SQL requests to the host database. It does this on the basis of a mapping between a conceptual schema, like ABCD, and the host's internal database schema in real time.

[9] http://ww3.bgbm.org/protocolwiki/
[10] http://chm.grinfo.net/index.php
[11] http://earth.google.com/kml/
[12] http://speciesanalyst.net/docs/dwc/index.html
[13] http://www.bgbm.org/tdwg/CODATA/Schema/default.htm

1.9.DiGIR PHP Provider

The DiGIR PHP provider is by far the most common implementation of the DiGIR protocol. This software works on a similar basis to the PyWrapper. Incoming requests are converted into SQL queries against the hosts database.

1.10.DiGIR2 Provider

DiGIR2 is the working name of a new data provider package not a new version of the DiGIR protocol. It works on a radically different principle to the other providers (both DiGIR PHP and PyWrapper). It adopts the use of semantic web technologies. Data from the host database is mapped into an RDF triple store using a custom mapping language. The triple store is then exposed to the internet via a number of possible protocols and query languages. By default the provider is able to support the W3C standard SPARQL protocol and query language. There are plans to support LSID and possible DiGIR and TAPIR protocols.

1.11.LSID

Life Science Identifiers (LSID)[14] is a GUID standard that defines a resolution mechanism with bindings to different web service protocols. LSID is currently being evaluated by the GUID project[15] but is likely to be rolled out by some providers in the next year.

1.12.Other Schemas

There are a number of other schemas under development within TDWG that are not currently exchanged using any formal protocol. These include TCS (although SEEK have a demonstration SOAP web service), SDD, TaxMLit and TaxonX. A presentation was submitted to the meeting on TaxonX which can be seen on the meeting wiki.

2.Protocol outlook for 2007

The outlook for take up of the different protocols above can be summarized in a simple table:

Protocol	**Number at end 2007**
DiGIR	200+
BioCASe	~100
TAPIR	10 possibly 40+
SPARQL	30+
LSID	10?

It appears necessary for the biodiversity informatics community to move towards exchanging identifiable objects. Identifiable objects are necessary for data integration. The first steps have to be taken using the protocols discussed here. Changes to the protocols or providers have to be minor, and may take years to be rolled out. It is likely that possible solutions will involve a combination of innovative use of existing technologies and web services that wrap some data providers.

[14] http://xml.coverpages.org/lsid.html

[15] http://wiki.gbif.org/guidwiki/