8-1-2014

A Services' Frameworks And Support Services For Environmental Information Communities

Nicholas J. Car

Matthew Paul Stenson

Simon J. D. Cox

Robert A. Atkinson

Peter Fitch

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

A SERVICES FRAMEWORK AND SUPPORT SERVICES FOR ENVIRONMENTAL INFORMATION COMMUNITIES

N.J. CAR(1), J. YU(1), S.J.D. COX(1), M.P .STENSON(1), R. ATKINSON(1) & P. FITCH(1)
(1): CSIRO Land and Water in, Dutton Park Brisbane QLD, Highett Melbourne VIC, Lucas Heights NSW & Black Mountain Canberra ACT, Australia

For environmental datasets to be used effectively via the Internet, they must present standardized data and metadata services and link the two. The Open Geospatial Consortium's (OGC) web services (WFS, WMS, CSW etc.), have seen widespread use over many years however few organizations have deployed information architectures based solely on OGC standards for all their datasets. Collections of organizations within a thematically-based community certainly cannot realistically be expected to do so. To enable service use flexibility we present a services framework - a *Data Brokering Layer* (DBL). A DBL presents access to data and metadata services for datasets, and links between them, in a standardized manner based on Linked Data and Semantic Web principles. By specifying regular access methods to any data or metadata service relevant for a dataset, community organizers allow a wide range of services for use within their community. Additionally, a community service profile testing service – a *Conformance Service* – may be run that reveals the day-to-day status of all of a community's services to be known allowing both better end-user experiences and also that data providers' data is acceptable to a community and continues to remains available for use. We present DBL and Conformance Service designs as well as a whole-of-community architecture that facilitates the use of the two. We describe two Australian environmental community implementations: eReefs and Bioregional Assessments and plans for wider deployment.

INTRODUCTION AND BACKGROUND

Information Platforms (IP) bringing together a range of geospatial, service-delivered, data sources from distributed systems, perhaps located in multiple agencies, must provide a services index. To be really useful, IPs cannot just rely on listing standardized data services using standardized metadata for a number of reasons, some of which are that they:

- are only about data, not how to discover and use that data;
- do not cater for relating datasets to one another other than via basic parent/child relationships or simple spatial intersects;
- cannot easily represent multiple services, of different types, for a single dataset;
- are entirely geospatially oriented, not all datasets are;
- do not link in to other metadata systems such as vocabulary services well.

The main issue that underlies some of the shortcomings of using metadata services such as the Catalog Service for the Web (CSW)[1] delivering metadata according to schema such as

[1] Catalogue Service: http://www.opengeospatial.org/standards/cat

ISO19115[2] and ANZLIC as indexes of service-delivered datasets is that the application schema they implement are still, despite a decade or more of development, digital library cards. They contain information about the data within the dataset – who created it, when, licensing etc. – but they do not contain the kinds of functional metadata that humans and automated data collection systems need in order to efficiently access the data. This functional metadata includes service end points, information models used, definition terms, system response times and so on.

The digital library card-based design of these metadata schema also limits their ability to take advantage of multi-source web data integration as AJAX[3]-powered applications do. For example, the widely used implementation of a CSW service, GeoNetwork[4], is unable to calculate the bounding box or temporal extent of the data it describes at record delivery time ('on the fly') therefore it must store it statically which is not ideal when service-delivered datasets, and thus their spatial and temporal extents, can change rapidly.

There are a range of methods already employed to bring synchronicity to metadata record content and service-delivered data, such as Geoserver/GeoNetwork integration[5], but these methods are not standardized, only work with specific standards implementations, not the standards themselves, and do not cater for a wide range of service types.

Geospatial web services contain metadata retrievable on per-record and per-service bases , the latter via capabilities request (for OGC service, this is a *GetCapabilites* function call) however if multiple services are to be used in an IP including potentially non OGC services, and when library-style metadata about the service delivered data, such as licensing information, must be known, then just relying on this integrated metadata is not possible.

Here, we describe our implementation of a services' framework which aims to build on the library-style metadata records of traditional data indexes with metadata collected from web service calls and other dynamic web functionality. It does this by adding an abstraction layer to data and metadata services. We hope to provide a service delivering, in an organized way, a large range of services and metadata types and yet one simple enough for basic human use. It should also be able to be used with advance, automated, clients as the Semantic Web grows[1].

In addition to detailing our specific implementation of a services' framework, the eReefs Data Brokering Layer (DBL), we list the specific Use Cases that prompted its design, its current use and plans for its future use in cross-IP contexts. We also describe the eReefs Conformance Service (CS) which is both a client of, and an assistant service for, the eReefs DBL.

DESIGN REQUIREMENTS

In [3], categories of Use Cases for IPs delivering distributed data services are considered with the paper building on earlier IP design work [4]. Table 1 lists Use Cases in [3]'s 5 categories. By considering the categories of Use Cases in [3], IP designers are prompted to cater for system requirements beyond those normally articulated by end users, all of which tend to fall within category 1. Of interest regarding service and dataset indexes are, of course, Use Cases in the "End User" category, such as "Discover Data", but also those in the "Data and Functionality Provision". In addition to providing data discovery and access, an IP should cater for the addition of new data to a service and the additions of new data and metadata services. Given there are many popular but non-OGC services available to data users, it is reasonable that the

---

[2] ISO 19115-1:2014 Geographic info – Metadata & http://anzlic.org.au/policies_guidelines
[3] http://en.wikipedia.org/wiki/Ajax_(programming)
[4] http://geonetwork-opensource.org
[5] http://geoserver.org/display/GEOS/GeoServer+GeoNetwork+Integration &http://geonode.org

“Add a new service” Use Case contain a specialized form reading “Add a new *type of* service”. This then requires that an IP supply protocols for doing so as it will be an unknown entity.

Table 1: Use Cases listed according to categories recommended in [3]

Title	Category
Discover Data	1. End User
Access Data	1. End User
Add new data to a service	2. Data and Functionality Provision
Add a new service	2. Data and Functionality Provision
List datasets not compliant with the IP data model	3. Enablement and Governance
Link vocabulary terms to external vocabularies	4. Cross-business Domain Integration
List all services not responding in a timely manner	5. System Maintenance

A SERVICES’ FRAMEWORK

We have chosen to consider metadata about service-delivered data as one *view* of a dataset. Similarly, the data itself is another *view* of that dataset. If the same dataset were delivered in multiple formats via multiple services, they would all be different *views* of the same dataset. If multiple formats/community schema implementations of metadata were delivered for a dataset, they too would all be different *views* of that dataset. This then places the conceptual dataset entity outside specific implementations of data and metadata and allows for an infinite set of *views* of to be related, through linking via the conceptual entity, to each other. *Views* related to the dataset that are neither data or traditional metadata can be thought of too with an example being a *provenance* view which describes the processes that lead to the creation of a dataset. Figure 1A shows a graphical representation of this concept.

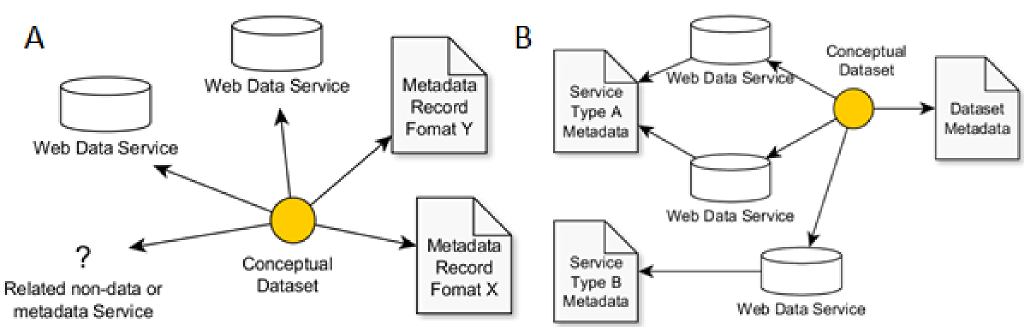

Figure 1: A, conceptual dataset with a series of data, metadata and other views and B, conceptual dataset linking to both dataset metadata and data services metadata

To implement such an information model, a mechanism must allow for the generation of abstract identifiers to identify the conceptual dataset and then allow linking to an unknown number of *view* realizations. In order to be really useful, a mechanism must also be able to deliver metadata about the services or data realizing each *view*. This is in contrast to the metadata about the dataset itself. Finally, the implementing mechanism must be based on a very flexible information model that allows for a large, and expandable, set of allowed *view* types.

Implementing a *view*-based index, or services’ framework service, would not prevent an IP from also implementing current metadata catalogue services, or current data services or even linked data and metadata services. The services’ framework service would simply list existing services as *views* of conceptual datasets. Preserving existing data/metadata service links is easy since the purpose of the *view*-based approach is to create such links.

By allowing an unbound set of *views* for datasets, an IP caters for the Use Case of "Add a new service", in fact it allows for services to be added to existing datasets as well as new services for new datasets which would require a new conceptual dataset entity to be created.

In order to allow for the "Add a new *type of* service" Use Case, a services' framework service implementation must either allow service metadata to be defined elsewhere and linked to or allow for the storage of new service metadata. With case, the former approach will prove more useful for, compared with the latter, it will reduce the burden on the IP designers in freeing them from having to implement a services metadata system and also allow service contributors to supply metadata for services from elsewhere, such as service developers' resources. This further linking from a conceptual dataset's *views*, realized as services, to metadata about those services, creates a graph which Figure 1B depicts.

Semantic Web Implementation of a Service Framework

A radically different way to describe data, data services and dataset relations compared with strongly constrained data model-based methods is presented by the use of the Sematic Web [1]. The Semantic Web promise is that people and automated agents can use a "Follow-Your-Nose" methodology [2, ch.4] to incrementally discover more data and metadata about things by following typed links from a starting point. Linking an ontology defining concepts in a certain area to other ontologies allow a client to traverse knowledge domains. This allows domains to be integrated which is an imperative for situations, such as that of web data services and metadata indexing, where lots of work has taken place to define concepts in disparate areas. It would be unrealistic to create a new, single model for all these domains.

By employing generic Internet and Semantic Web standards such as the Hypertext Transfer Protocol (HTTP)[6] and the Resource Description Framework (RDF)[7], graphs of any related objects can be made. The use of such standards then allows for a Services' Framework graphs constructed around nodes of existing data and metadata items. URI[8]-based information resource ID can be created ('minted') for each conceptual dataset and then data and metadata views linked to it via links of relevant types defined by the Services' Framework Service. Library/archive style metadata ('metadata about data') could be delivered at points in a conceptual dataset's graph, service details in another and metadata about how to use the services such as service entry points, data models and versions ('service metadata') could be delivered one further edge away. People or automated agents wising to use a dataset's service, may not need to traverse this second edge if they are already familiar with the data service.

As new data services become accepted for use, provided they have service metadata defined elsewhere, all an IP designer would need to do to allow for their implementation would be to add a new link type to the Services' Framework.

Collections of Datasets – Data Provider Nodes

IPs connect multiple, service-delivered, data sources from distributed systems together or at least present access to them in one place. In order to allow services from multiple owners to be used in multiple IPs, we implement a nodal structure whereby services owners establish a *Data Provider Node* (DPN) that each present an index of *Datasets* (the conceptual datasets in the preceding Figures). Figure 3 shows such a structure. With DPNs delivering their data service

---

[6] Hypertext Transfer Protocol: http://www.w3.org/Protocols/
[7] Resource Description Framework: http://www.w3.org/RDF
[8] Uniform Resource Identifier: http://en.wikipedia.org/wiki/Uniform_resource_identifier

indexes in a consistent, expandable, machine navigable way, an IP can generate a master services index almost dynamically (it can crawl the indexes of the various DPNs within its purview regularly) and present it in the same manner as the DPNs do. This allows further harvesting of service indexes by other IPs which can treat this IP exactly like a DPN.

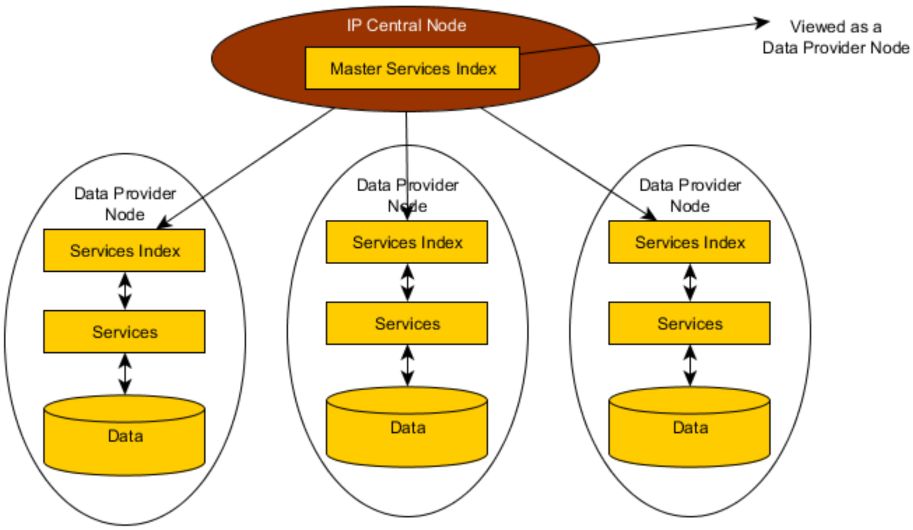

Figure 2. A generic, nodal, information platform architecture showing service indexing

DESIGN SPECIFICS

Data Brokering Layer

Our implementation of a services' framework, the eReefs Data Brokering Layer, describes services and functional metadata for them. Functional metadata is metadata that can be used by machine clients to decide what tasks to carry out on the data. Metadata of, say service type and service endpoint (a URI) tell a client where to find the service and what function calls will work when there. Our description tool is an OWL[9] ontology called the Data Provider Node Ontology (http://purl.org/dpn). It contains classes for *Data Provider Nodes* (DPNs – institutions or sub-institutional groups) that contain *Services* which have various *Interfaces*. The list of described service types can be added to allowing IPs to deliver a growing range of services. The use of an ontology allows conceptual dataset entities (URIs) to act as an index to which services are related via links. Figure 4 gives pseudo code in *turtle*[10] format for RDF data for Figure 1A.

The class *dpn:CSW* in Figure 4 is detailed in the DPN ontology from which we learn it is a Catalog Service for the Web, it is standardized by the OGC and that testing data for the particular service version is available. For CSW 2.0.2, as in the example in Figure 4, the test suite is at http://cite.opengeospatial.org/teamengine/about/csw/2.0.2/web/.

The DBL also contains a persistent ID layer which abstracts the public addresses of Data Provider Nodes, Datasets and services from implementation locations. Our implementation of is the PID Service[11] that operates as an advanced Apache web server's *mod_rewrite*[12] with pattern match inheritance and lookup function abilities. Such a layer fulfills the requirement of

[9] OWL, the Web Ontology Language: http://en.wikipedia.org/wiki/Web_Ontology_Language
[10] Turtle - Terse RDF Triple Language: http://www.w3.org/TeamSubmission/turtle
[11] Persistent Identifier Service: https://www.seegrid.csiro.au/wiki/Siss/PIDService
[12] Apache mod_rewrite homepage: http://httpd.apache.org/docs/current/mod/mod_rewrite.html

allowing for the creation of abstract dataset identifiers and the Linked Data reserved URI register /datasets/ is used to contain them. It also allows for the simple linking of conceptual datasets to any number of *views* by the use of a *_view* Query String Argument appended to a dataset's base URI. Using the dataset URI from Figure 4, the URI [http://example-dpn.org/dataset/abc123def456?**_view=dpn:CWS**](http://example-dpn.org/dataset/abc123def456?_view=dpn:CWS) redirects to http://example-dpn.org/dataset/abc123def456/service/MetadataServiceX since MetadataServiceX implements the CSW standard. This redirect need not redirect to a location within the DPN's domain thus allowing DPNs to act as registries of services, as well as repositories of them.

```
<http://example-dpn.org/dataset/abc123def456>
  a                   dpn:Dataset;
  dcterms:title       "Ocean Colour"^^xsd:string;
  dpn:service         :MetadataServiceX;
  dpn:service         :MetadataServiceY;
  dpn:service         :WebDataServiceA;
  dpn:service         :WebDataServiceB;
  dpn:service         :OtherService;
.

:WebDataServiceA
  a                          dpn:Service;
  dpn:hasServiceType         dpn:DataServiceInterface;
  dpn:hasServiceInterface    dpn:WCS;
  dpn:hasImplementation      dpn:THREDDS;
  dpn:serviceEndpoint        "http://thredds0.nci.org.au/thredds"^^xsd:anyURI;
  dpn:catalogEndpoint        "http://thredds0.nci.org.au/thredds/catalog.xml"
                                                            ^^xsd:anyURI;
.

:MetadataServiceX
  a                              dpn:Service;
  dpn:hasServiceType             dpn:MetadataServiceInterface;
  dpn:hasServiceInterface        dpn:CSW;
  dpn:hasImplementation          dpn:GeoNetwork;
  dpn:hasImplementationVersion   "2.0.2"^^xsd:string;;
  dpn:serviceEndpoint            "http://aodaac1-mel.vic.csiro.au:8080/geonetwork/srv/en/csw"^^xsd:anyURI;
```

Figure 3: Pseudo RDF code in turtle representing part of the conceptual dataset in Figure 1A

This combination of a functional ontology and a persistent ID/redirection layer allow DPNs to locate datasets and services anywhere they like using a myriad of implementation methods and yet still present access to them in a consistent and persistent manner.

Conformance Service

Since standardized access to DPN's services is guaranteed through adherence to DBL requirements, a Conformance Service (CS) can run that operates as a web crawler testing parts of DPN's services for valid responses. The CS runs regularly allowing the eReefs community to know the status of all its services, regardless of which DPN they are in, and individual DPN owners the status of their services as seen by the eReefs IP. It is possible that services may be checked by a number of CSes in which case the DPN owner may receive multiple reports.

The CS itself is based on a network monitoring service, Nagios[13], but with modules allowing Semantic Web [1] queries and inferencing. The CS can "follow its nose" to enter a DPN, discover what services it offers and discover metadata for them without having a complete schema for the DPN. This ensures it is always able to pick up changes in DPNs, such as service addition, removal or expansion, without requiring notification messages.

Table 2: A partial list of the Conformance Service's tests (from [14])

Name	Description	Result
DPN Online	Does the DPN's base URI resolve in human-readable and machine readable format?	HTML & RDF resources at the DPN's base URI
Dataset Index found	Can the DPN's dataset index be found (at http://{DPN_BASE_URI}/dataset/) and does it resolve in human-readable and machine readable formats?	HTML & RDF resources at DPN's datasets URI

[13] http://www.nagios.org

[14] eReefs design wiki CF page: https://wiki.csiro.au/display/ER/Barry+-+Conformance+Service

Dataset index traversal	Can the DPN's dataset index in RDF be traversed with entries being of type dpn:Dataset and does each entry meet minimum dpn:Dataset requirements?	true or false for each dataset
Individual dataset inspection	Does a dataset's base URI resolve, is its list of services discoverable, are each of those services described as being of a known type (e.g. dpn:WebFeatureService)	true or false for each dataset aspect
Individual service inspection	Does each service for a given dataset have all the attributes required of its type, is its end point returning the correct response, is it reporting in a timely fashion	true or false for each service aspect
Service implementation compliance	Does the reported standardized service pass test suites supplied by its own specification (i.e. is it configured correctly according to its spec)	Vendor test suite results wrapped in CS functions

A testing service eases, rather than inhibits, the addition of new IP services. In facilitating Use Cases in the "Data and Functionality Provision" category (see Table 1), the CS provides an coaching method for DPN owners who wish to add new services, or new types of service, to an IP. New services can be added and the CS will report where misconfigurations occur. If the CS is run daily, a report of their new service's conformance to the IP's expectations will be issued to the DPN owner every day, perhaps with the service remaining unpublished until it is fully conformant. The service is similar in concept to Google's Webmaster Tools[15] for websites. Table 2 lists some conformance tests that are applied to every DPN.

When testing a standardized service version, say the OGC's CWS 2.0.2 as per the example in Figure 4, the CS will, if able, use testing functions supplied by the standardized service's specification. In this example this entails running the test suite is at http://cite.opengeospatial.org/teamengine/about/csw/2.0.2/web/. The CS would run functions from the CSW specification, such as *GetCapabilities* and *GetRecordById* checking for correct results. To do this, the CS needs to be wrap service test suites in its own functions to apply them at will.

For a new type of service to be added to an IP, a test suite that checks for instances of that service type's correct configuration need to be supplied to the IP owner for inclusion in the CS.

IMPLEMENTATIONS

Two very different IPs being developed in Australia use the architecture and principles here as a basis: the eReefs IP [5] and the Bioregional Assessments IP (BAIP) [6]. eReefs is a distributed collection of services from a number of Australian government agencies aiming to create a near-real-time view of water quality in the Great Barrier Reef's lagoons whereas the BAIP sources a lot of data not from services but from a digital repository. While it does contain standardized services too, the BAIP extends the requirements for dataset representation to unstructured 'bins' of data that cannot have nearly as much functional metadata associated with them as standardized web services do since there is no guaranteed pattern of access.

Since eReefs is a thematic IP about water quality, all of its services relate to some aspect of it. A vocabulary service has been established that lists water quality terms and the relations between them encountered in eReefs' primary datasets. Linking to these vocabulary terms from datasets occurs at a sub-dataset level (within the data) however a vocabulary term use *view* is now being tested for each dataset listing all the terms its data references.

Since the BAIP cannot rely on a Conformance Service that checks each dataset according to the specification of the standard it implements since many of its datasets are not service-

[15] https://www.google.com/webmasters/tools/home?hl=en

delivered, it uses its own project ontology in addition to the DPN ontology to describe datasets. Its ontology is a specialization of a Provenance ontology, PROV-O, [7] that allows various provenance-related *views* of datasets to be created. In the BAIP's case then, one implementation of the "Related non data or metadata service" in Figure 1A is a provenance service.

FUTURE USE

The eReefs and BA IPs are pathfinder projects for the Australian National Plan for Environmental Information's National Environmental Information Infrastructure (NEII). This project's goal is to "enhance discovery of, and access to, national priority environmental information". These authors' future vision for the NEII is that institutions and interest groups across Australia will be able to build DPNs delivering services that any NEII-conformant IP can index. It is conceivable that a single, national IP be established listing all the services in all IPs in Australia however the computational load of implementing CFs at that level is not yet know.

REFERENCES

[1] Berners-Lee, T. (2001). The Semantic Web. Scientific American. Online at http://www.scientificamerican.com/article/the-semantic-web/, accessed 2014-03-01.

[2] L. Yu, A Developer's Guide to the Semantic Web. DOI 10.1007/978-3-642-15970-1_14, Springer-Verlag Berlin Heidelberg 2011.

[3] Car, N.J. and Murray, N., 2013. Implementing and adapting the WRON-RM Use Case categories for eReefs. MODSIM2013, 20th International Congress on Modelling and Simulation. MSSANZ, December 2013, pp. 2124- 2130. ISBN: 978-0-9872143-3-1. Online at http://mssanz.org.au/ modsim2013/K3/car.pdf, accessed 2014-02-25.

[4] Lemon D, Cox S, Walker G, Atkinson R, Fitch P and O'Hagan R. 2007. Water Resources Observation Network – Reference Model Version 0.1. CSIRO: Water for a Healthy Country. 93 pp. Online at: http://www.clw.csiro.au/publications/waterforahealthycountry/2007/ wfhc-WRON-Reference-Model-V0.1.pdf, accessed 2014-02-25.

[5] eReefs Project Management (2014), eReefs project homepage: http://ereefs.org.au, accessed at 2014-03-23.

[6] CSIRO (2014), Bioregional Assessments. Project homepage: http://www.csiro.au/Organisation-Structure/Flagships/Water-for-a-Healthy-Country-Flagship/Water-Resource-Assessment/Bioregional-Assessments.aspx, accessed 2014-03-23.

[7] Lebo, T., Sahoo, S. and McGuinness D. (eds.) (2013) PROV-O: The PROV Ontology. W3C Recommendation 30 April 2013. http://www.w3.org/TR/prov-o, accessed at 2014-03-23.