USGIN information exchanges

# Introduction

A USGIN information exchange is a specification for how to expose a particular kind of data through a particular service protocol in a particular interchange format. The specification includes 1) an abstract content model for the information that defines the entity or entities of interest, their properties, and the logical data type to specify property values, possibly including controlled vocabularies; 2) and implementation of the content model using a specific encoding scheme to define the information exchange format; 3) the service protocol or protocols that will be used for accessing information. The development of information exchange specifications outlined here is based on operational systems of the Open Geospatial Consortium (OGC, http://www.opengeospatial.org/ogc/process), International Organisation for Standardisation (ISO, http://www.iso.org/iso/home/standards\_development.htm), and the EPA Environmental Information Exchange Network (<http://www.exchangenetwork.net/data-exchange/>), and the National Geothermal Data system (NGDS, http://geothermaldata.org/page/data-interchange-content-models).

# Content model

Content models are abstract specifications that define a feature for information interchange, along with the properties associated with the feature and relationships to other features. The content model in its simplest form is a statement that some particular entity (feature or observation) will have some list of properties. For example a content model for a record describing a book might include: title, author, publisher, publication date, publication place, number of pages, and an ISBN number (a URI).

New content models are developed based on community needs. A network participant may propose a new model for development and form a working group to develop a draft for review and potential adoption. Content models may be developed as Excel workbooks, text descriptions, or UML models. Documents used in the development process are hosted in a version-controlled (VC) repository. Documents defining each model are managed in one repository. The development documents describe and specify the model, but do not implement the model in a computer-processable form. These documents are intended to help human users understand the content model.

Review copies of model specification documents are made available at a web location (typically not the development version-controlled repository) with access control based on the target reviewer community. Availability of a model for review is publicized in a request for comment broadcast to the intended user community, with instructions for commenting, and a time frame during which comments will be accepted. Comments are collected by e-mail and compiled in a comments spreadsheet that is maintained in the content model VC repository. At the end of the review period, the workgroup incorporates comments to produce a final specification document. When the group is satisfied that the model is ready for implementation and utilization, a copy of the specification document is moved to an archival ‘tag’ repository, and locked against further modification.

A URI is assigned to the model at this point; the specification document is considered the normative representation of the model. Ideally the model URI will dereference to the specification document. The URI should include version information (see versioning and URI’s section, below).

# Interchange format

In order to actually use a content model, it must be implemented using an encoding scheme that is computer processable. The eXtensible Markup Language (XML) is typically used to implement the content models, but other schemes (e.g. JSON) may be adopted as technology evolves. The important thing is that the information is encoded consistently in a structured, well documented format. This enables computer programs to parse the interchange documents and extract desired information, and as newer encodings are adopted, conversion between formats can be automated using software.

Test implementation of a content model should be done during the review of the abstract model, because this provides an excellent review of the model. When the specification is finalized (‘tagged’), a final implementation can be developed. The implementation is specified by a separate document. In the case of XML implementations, the normative specification is an XML schema; other encoding schemes will have other normative implementation specifications. The implementation encoding is identified by a URI (a Namespace URI in the case of an XML implementation). This URI should be included in any instance document based on that interchange format implementation such that software consumers of the document content can determine that the document is encoded using a known format.

Each implementation must define a validation process that will determine if an instance document is conformant with the implementation.

# Service Protocol

# Service Contract

# Versioning and URIs

Tagged model and implementation specifications are assigned USGIN URIs to identify them. The identity of a model is defined by the collection of feature types, properties, cardinalities, data types, and relationships. The identity of an implementation is defined by the collection of all instance documents that are valid according to the validation processes defined by that implementation.

# Glossary

conformant – an artifact (model, implementation, instance document) implements all normative provisions of a specification exactly. The artifact may implement other features that are not normative in the specification (http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/48\_conformance.png

compliant--the system provides support for some of a given standard

instance document: a document containing information of interest encoded according to some interchange format specification.