ISO TC211 Meeting Riyadh 2006

Encoding rules ISO 19118 – Revision 1

Claude Eisenhut
ce@eisenhutinformatik.ch

Agenda

- Overview of work done
- Discussion of features of encoding rules
- Next steps

Work program proposed in Orlando (2006-05)

- comparison document (UML, XMI 2.0, 19136, 19139) at XSD level. May need to consider instance level as well.
- collect the differences
- based on the list of differences, discuss and collect explanations/standpoints/opinions
- formulate priniciples and clarify them in 19118

Status

- comparison table created and sent to nominated experts (5-Nov-2006)
- Asked experts about:
  - corrections
  - common features among these encoding rules
  - unique features of each one of these encoding rules
  - how much flexibility in schema production is required? What is the user benefit of this flexibility?
  - what is the role of XML Schema (at design/construction time, at runtime)?
  - what is the role of UML (at design/construction time, at runtime)?
- so far no comments received

Features of encoding rules (to be discussed)

- Object, property, object pattern
- Base xml schema type
- DataTypes
- Instance extensibility
- Use of any existing xml schemas
- Multiple inheritance
- Ordering of properties
- Ordering of values
- Nested objects/remote properties
- Kind of association
- CodeLists
- Serialize properties
- Both ends of an association are navigable
- Types from harmonized model
- Extending the content model
- Other model elements

Features of encoding rules (discuss if time left)

- Names
- Common (technical) attributes
- IDs
- Documentation
- Mapping to schema and schema documents
- Use of xs:extension
- Specialized attributes
- Metadata properties
- XSD details

Object, property, object pattern

- object/property/object pattern (19136/19139)
- use of xsi:type (XMI)
- encode properties as xsd:elements and/or xsd:attributes (XMI)

Base xml schema type

- base type (19136/19139)
- no base type (XMI)
- Feature types use different base types than ordinary classes (19136)

DataTypes

- instances of DataTypes don't have an identity (19136/XMI)
- instances of DataTypes have an identity (19139)
- instance production of DataType is different from schema production (XMI)

Instance extensibility

- extensibility for every kind of object (XMI)

Use of any existing xml schemas

- make use of any existing xsd:types (19139/XMI)

Multiple inheritance

- support multiple inheritance (XMI)

- *Should 19103 disallow it, if it is not supported by the encoding rules?*

Ordering of properties

- ordering/no ordering of properties (XMI)
- flexible ordering of properties (19136)

Ordering of values

- no expression if values are ordered (19136/19139/XMI)
- *How gets an application aware if ordering is relevant?*

Nested objects/remote properties

- no flexibility if objects inline or by reference (19139/XMI)
- flexibility if objects inline or by reference (19136)
- different instance if target object is not in same document (XMI)

Kind of association

- special schema production for collections (19136)
- different schema if association/aggregation/composition (19136/XMI)
- same schema if association/aggregation/composition (19139)

CodeLists

- No schema generation (19136/19139)
- flexibility in encoding properties of type CodeList (19136)

- *How get’s an application aware of the to be used CodeLists?*

Serialize properties

- (don't) serialize (derived) properties (XMI)

- *What do 19136/19139 with derived properties?*

Both end of an association are navigable

- express reverse property in xsd (19136)

ce@eisenhutinformatik.ch

Folie 19

Types from harmonized model

- encoding of types from harmonized model is according to general encoding rules (19139/XMI)
- encoding of types from harmonized model is fixed and not according to general encoding rules (19136)

ce@eisenhutinformatik.ch

Folie 20

Extending the content model

- add new class (19139)
- extend existing class (19139)
- extend existing CodeList (19139)

- *Breaks interoperability with other encoding rules!*

Other model elements

- association class not supported (19136/19139/XMI)
- n-ary association not supported (19136/19139/XMI)
- class scope properties (XMI)
- no defined handling of other model elements (19139/XMI)
- other model elements are ignored (19136)

- *Should 19103 disallow it, if it is not supported by the encoding rules?*

Names

- **name of xs:type/xs:element may not be different from uml:name (19136/19139)**
- **name of xs:type/xs:element may be different from uml:name (XMI)**

Common (technical) attributes

- common attributes for every kind of object (19136/19139/XMI)

IDs

- name of id xml:attribute can not be changed (19136/19139)
- name of id xml:attribute may be changed (XMI)
- oid is a well known xml:attribute (19136/19139)
- oid may be a property instead of an xml:attribute (XMI)

Documentation

- forward documentation to xsd (19136)
- documentation is ignored (19139/XMI)

Mapping to schema and schema documents

- flexibel namespace and ns-prefix (19136/19139/XMI)
- split a schema into different schema documents (19136/19139)

Use of xs:extension

- make use of schema extension (19136/19139)
- make use of schema extension or copy down properties (XMI)

Specialized attributes

- allow specialized attributes, but ignore it for xsd generation (19136)

Metadata properties

- metadata tagged in xsd (19136)

XSD details

- set xsd:form (attribute) (XMI)
- set xsd:nillable (element) (XMI)
- set xsd:default (attribute) (XMI)
- set xsd:fixed (attribute) (XMI)

Next steps

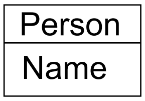
- get a CD at end of january based on the following principles
  - keep flexibility level low
  - keep compatibility with existing GML instances
  - keep compatibility with exisiting Metadata instances

What are encoding rules? (generally)

- they enable to separate system implementation considerations from the specification of the content (geographic information)
- but if the content is not exactly defined, there is no interoperability between different encodings

What are encoding rules? (data transfer)

- prescribe, how to encode data (considering the engineering viewpoint) according to it‘s model / content description (the information viewpoint)
- Example for XML: xml-elements or xml-attributes?



```
<Person>
  <Name>Eisenhut</Name>
</Person>
```

OR

```
<Person Name=„Eisenhut“>
</Person>
```

Why do the same encoding rules matter? Reuse!

- One encoder/decoder for any kind of data (features, feature catalogs, crs, registries, metadata, models, ...)
- One web service for access to any kind of data (Do we really need catalog service and web feature service?)
- Enables to raise the level of xml-parsing from elements to objects (see also XMI and RDF parsers)
- Do we have the ressources to develop multiple times similar specifications?

The technolgy viewpoint

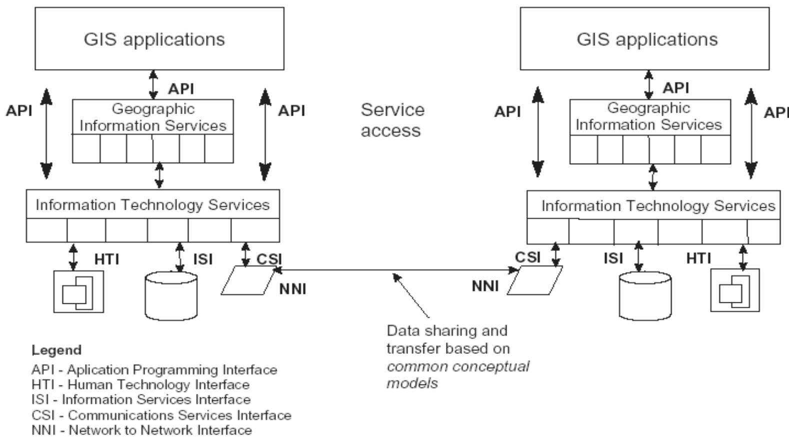

GIS applications
API
Geographic Information Services
API
API
Information Technology Services
HTI
ISI
CSI
NNI
Service access
Data sharing and transfer based on *common conceptual models*
GIS applications
API
Geographic Information Services
API
API
Information Technology Services
CSI
ISI
HTI
NNI
Legend
API - Aplication Programming Interface
HTI - Human Technology Interface
ISI - Information Services Interface
CSI - Communications Services Interface
NNI - Network to Network Interface

Figure 12 — The Architectural reference model

Source: ISO 19101