

Validación de ficheros XML de metadatos de información geográfica. Norma ISO19139

Amaro, Alberto⁽¹⁾

(1) Área de Teledetección. Departamento de Observación de la Tierra
INTA
Ctra. Ajalvir sn Torrejón de Ardoz Madrid
915201656, 915201945,amaroca@inta.es

Resumen

La versión definitiva de la norma ISO19139 ha sido publicada el 17 de abril de 2007. En la página web de la ISO es posible descargar el conjunto de ficheros creados siguiendo la especificación del W3C para esquemas de XML. Esta es la implementación del esquema necesario para validar la apariencia de los ficheros de metadatos de información geográfica. Su aplicación definitiva permitirá unificar la apariencia de los ficheros XML y, con ello, impulsar la interoperabilidad entre todas las fuentes de información.

El esquema ha sido creado por la organización EDEN (Equipo de Expertos en Normalización) del IGN francés.

El objetivo de esta ponencia es presentar los progresos y dificultades encontrados durante la aplicación de los esquemas definidos en la norma ISO19139. Por un lado se trata un caso práctico como ha sido la validación de los metadatos generados por el Área de Teledetección del INTA. Por otro lado, se ofrece una visión completa del esquema como consecuencia de su incorporación en la herramienta de validación de ficheros XML que incluye el software IME (ISO Metadata Editor). Este segundo caso ha incrementado la complejidad debido a que el objetivo ha sido abarcar todos los metadatos definidos en la norma ISO19115, pero al mismo tiempo ha permitido profundizar en el comportamiento de alguna de las entidades y elementos más heterogéneos.

El concepto fundamental para definir las dificultades encontradas se puede resumir en una frase: "es muy sencillo generar ficheros XML completamente validados por la ISO19139 pero que, sin embargo, no cumplan la normativa ISO19115". Evidentemente una contradicción que es tan cierta como justificada cuando se profundiza en los requerimientos de la norma ISO19115, y que debe ser tenida en cuenta si no se quiere correr el riesgo de reducir notablemente la interoperabilidad. En definitiva, el esquema ISO19139 no puede ofrecer un mecanismo de control absoluto sobre la calidad y validez de los metadatos.

En esta presentación, primero se muestra la arquitectura y filosofía de creación del esquema, así como el origen y estado actual del mismo. Posteriormente se enumeran algunas de las condiciones de la ISO19115 que, según los propios autores del esquema, no ha sido posible contemplar.

También se explican los errores más usuales que puede encontrar un usuario en el momento de validar sus metadatos en XML y se ofrecen soluciones. Por último se comentarán algunas propuestas para intentar reconducir la complejidad de los ficheros XML con la intención de valorar su aplicabilidad en un caso como el de metadatos de información geoespacial.

La herramienta de edición de metadatos IME permite verificar los ficheros XML a partir de

cualquier versión del esquema. Aprovechando esta opción se muestra un caso práctico de validación de datos en XML y se comentan algunos de los cambios más significativos que se han producido en la versión actual, disponible en la web de la ISO. También se muestra la utilidad de IME para conocer, a partir del esquema, la apariencia en formato XML de un elemento de la ISO19115, y para corregir y depurar el perfil en función de los errores que muestra la herramienta de validación.

Palabras clave: JIDEE 07, metadatos, XML, esquema, ISO19139, ISO19115

1 Introducción

La creación del esquema XML para validar los metadatos de información geográfica recogidos en la norma ISO19115 [1] ha sido llevada a cabo por el grupo EDEN del IGN de Francia [2] en un proceso que ha durado varios años. Se han ido generando diferentes versiones del esquema y, en el momento de escribir esta ponencia, ya se encuentran disponibles: una primera versión (identificada como "*19139 XSchemas TS RC 2006 May 4*") en la página web de esquemas de la ISO [3] y el documento final de la norma.

En la norma ISO19139 [4] de implementación del esquema se recogen las reglas para generar un esquema en lenguaje XML que permita validar ficheros XML con metadatos de la ISO19115. La intención de crear esta norma es, en la línea de la interoperabilidad, ajustar el amplio rango de posibilidades que se ofrecen para plasmar en un medio digital los metadatos de información geográfica. Limitar las opciones de codificación a un formato XML y fijar unas reglas del esquema que definan sus relaciones, los tipos de datos permitidos, sus restricciones, etcétera.

Por lo tanto, la norma ISO19139, tal y como se recoge en su introducción, es un medio para mejorar la interoperabilidad por medio de una especificación común para la descripción, validación e intercambio de metadatos relacionados con conjuntos de datos geográficos, series, fenómenos y atributos.

Como resultado de la complejidad de la norma ISO19115, el esquema en lenguaje XML que lo representa, no ha podido evitar mantener e incluso incrementar las dificultades, tanto para su creación como para su utilización por parte de los usuarios. En este artículo se tiene la intención de comentar algunos de estos aspectos.

En base a la experiencia acumulada en el Área de Teledetección del INTA durante varios años, con borradores previos del esquema y con la última versión disponible, se realiza una breve revisión de cómo se ha concebido el esquema, y posteriormente se comentan algunos de los puntos más destacados. El trabajo realizado en el Área de Teledetección ha estado apoyado por la herramienta IME [5] (ISO Metadata Editor del INTA) que se creó en 2003 para resolver el acceso a la norma ISO19115, la generación de ficheros XML y perfiles de metadatos y la validación de los mismos según el esquema y, actualmente, se encuentra en la versión 4.0.

2 Construcción del esquema de la ISO19139

El proceso seguido consiste en la representación de los metadatos geográficos definidos en la norma

ISO19115 como un conjunto de paquetes UML “Unified Modelling Language”. UML es un lenguaje de esquema conceptual que durante años se ha utilizado como paradigma de la representación de la programación orientada a objetos. En el caso del grupo de trabajo TC211 de la ISO, la definición del esquema conceptual del lenguaje se encuentra en la norma ISO19103 [6]

Los paquetes UML contienen una o más clases UML que representan las entidades definidas en la ISO19115 y también definen estereotipos (mecanismos de extensión), reglas de relación entre los elementos y condiciones.

La norma ISO19118 [7] define un conjunto de reglas de codificación para la transformación del esquema conceptual, construido en UML, en un esquema XML.

Según se indica en la norma ISO19139, la meta principal de la creación del esquema era mantener la interoperabilidad con otras especificaciones de la serie ISO19100, la extensibilidad y reusabilidad. Es decir, este esquema está basado directamente en otros esquemas del grupo de normas ISO19100 y mantiene la coherencia con las estructuras definidas en ellas. Reutiliza implementaciones que ya existen, como por ejemplo el esquema de GML [8].

2.1 Extensiones a los modelos UML

Ya se ha comentado que la norma ISO19115 define los metadatos de información geográfica en una versión independiente de la codificación que se vaya a utilizar. Es la norma ISO19139 la encargada de convertir esa jerarquía de metadatos en una codificación concreta, en este caso en un esquema XML que validará los documentos XML de metadatos.

Para llevar a cabo esta tarea y, en base a las reglas que define la ISO19115, ha sido necesaria la definición de nuevas extensiones a los modelos UML que ya existían en el conjunto de especificaciones ISO19100. Se han definido nuevos modelos para facilitar la interoperabilidad, el uso, la adaptación a los entornos web, etcétera.

Como ejemplo: la clase “*CharacterString*” definida en la ISO19103 ha sido extendida para cubrir requerimientos de su uso en entornos web. Y se han creado tres subclases:

Anchor (para poder usar hiperenlaces hacia URI's en *RS_Identifier* y *MD_Identifier*)

MimeType (para poder definir un tipo mime en *MD_BrowseGraphic*)

Filename (referencia a la URI de un fichero externo para *MD_BrowseGraphic*)

También se han definido extensiones para cubrir los requerimientos del multilinguaje.

La clase *PT_FreeText* amplía la capacidad del tipo de dato “*Free Text*” para permitir la escritura del valor de un elemento en diferentes idiomas.

Hay extensiones para definir las transferencias de información geoespacial basadas en metadatos. Se definen Conjuntos de Datos (*DS_Dataset*) como parte de Agregados (*DS_Aggregate*) que a su vez pueden ser subconjuntos o superconjuntos de otros Agregados.

Tanto los Conjuntos de Datos como los Agregados se componen de conjuntos de elementos *MD_Metadata* que se corresponden con la norma ISO19115.

Y aparecen nuevas extensiones para definir la organización del intercambio de esta información por transferencia: Conjunto de Datos de Transferencia (*MX_Dataset*) y Agregado de Transferencia (*MX_Aggregate*).

Si con *DS_Dataset* y *DS_Aggregate* es posible agrupar la información de diferentes conjuntos de

metadatos; con *MX_Dataset* y *MX_Aggregate* se define como va a ser esa transferencia (archivos de datos para conjuntos de datos, archivos de soporte con información sobre recursos relacionados, información complementaria, catálogos de sistemas de referencia, unidades de medida y codelists).

2.2 ¿Cómo se ha realizado la codificación de las reglas de la ISO19115 para convertirlas en el esquema de la ISO19139?

Recordemos que, principalmente, se han seguido las reglas de codificación especificadas en la ISO19118. Aunque esta norma comenta varios métodos de conversión entre UML y XML, hay una serie de reglas por defecto que se han seguido.

Durante el proceso de creación del esquema se fueron utilizando diferentes prefijos para los espacios de nombres. Finalmente se optó por los siguientes:

- *gco* (Geographic COmmon extensible markup language)
- *gmd* (Geographic MetaData extensible markup language)
- *gml* (Geographic Markup Language)
- *gmx* (Geographic Metadata XML Schema)
- *gss* (Geographic Spatial Schema extensible markup language)
- *gsr* (Geographic Spatial Referencing extensible markup language)
- *gts* (Geographic Temporal Schema extensible markup language)
- *xlink* (xml link language)

Los espacios de nombres se utilizan en los esquemas XML para facilitar la identificación unívoca de los elementos definidos en un esquema. Permiten evitar la duplicidad de nombres, pues el prefijo del espacio de nombres los convierte en un vocabulario distinto. La norma ISO19139 ha agrupado los elementos en “diccionarios”, por medio de estos espacios de nombres. Así por ejemplo, los elementos relacionados con metadatos de información temporal pertenecerán a un mismo espacio de nombres: “*gts*” y los elementos definidos en el GML pertenecerán al espacio de nombres “*gml*”. Es por lo tanto muy importante prestar atención al espacio de nombres de cada elemento porque en ocasiones los nombre “*no cualificados*” (sin el prefijo del espacio de nombres) coincidirán.

La Clase UML es el punto de partida. Una clase está definida por sus propiedades (representadas por pares: nombre-valor) y las propiedades pueden representar atributos o relaciones (asociación, agregación o composición).

Vamos a ver los pasos que se han seguido desde una clase definida en UML hasta su codificación en el lenguaje de esquemas:

2.2.1 Definición de un Tipo de Clase por Defecto.

- 1) Una Clase UML es codificada como un tipo de dato Complejo en el lenguaje XML. Un tipo de dato Complejo consiste en un dato formado por elementos y/o atributos. Los elementos pueden estar definidos como una secuencia (*sequence*), una selección única (*choice*) o una selección no ordenada (*all*).

Veamos el caso de la entidad *CI_Contact* definida en ISO19115 para dar información sobre un punto de contacto:

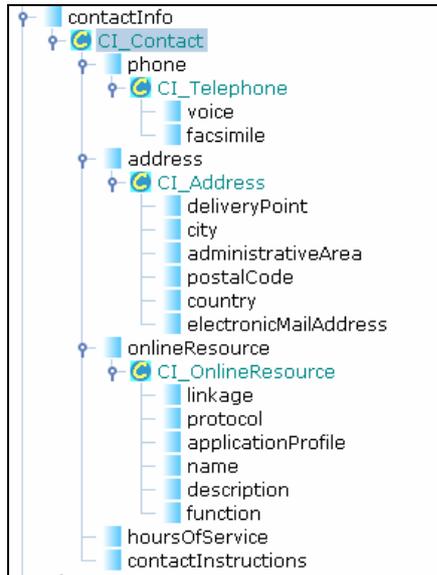


Figura 1. Entidad CI_Contact. Jerarquía de metadatos (imagen generada con IME 4.0)

Se define un tipo de dato Complejo. En la norma ISO19139 se identifican estos tipos de datos como XCT (XML Class Type) y en los ficheros del esquema siempre se añade al nombre de la entidad el sufijo “_Type”.

Así tenemos el tipo complejo CI_Contact_Type que se muestra en la figura 2.

```

<xs:complexType name="CI_Contact_Type">
  <xs:annotation>
    <xs:documentation>
      Information required enabling contact with the responsible person and/or organisation
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
  </xs:complexContent>
</xs:complexType>

```

Figura 2. Tipo complejo CI_Contact_Type (imagen generada con XMLSpy)

- 2) Estos tipos de datos Complejos contendrán comentarios (*xs:annotation* y *xs:documentation*) y un Contenido Complejo (*xs:complexContent*) que permite derivar un tipo de dato complejo a partir de otro (relación de herencia) o por restricción o por extensión de sus elementos y/o atributos. Si se expande completamente el árbol de la figura 2, tendremos el diagrama de la figura 3.

```

<xs:complexType name="CI_Contact_Type">
  <xs:annotation>
    <xs:documentation>Information required enabling contact with the responsible person and/or organisation</xs:docu
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
      <xs:sequence>
        <xs:element name="phone" type="gmd:CI_Telephone_PropertyType" minOccurs="0"/>
        <xs:element name="address" type="gmd:CI_Address_PropertyType" minOccurs="0"/>
        <xs:element name="onlineResource" type="gmd:CI_OnlineResource_PropertyType" minOccurs="0"/>
        <xs:element name="hoursOfService" type="gco:CharacterString_PropertyType" minOccurs="0"/>
        <xs:element name="contactInstructions" type="gco:CharacterString_PropertyType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Figura 3. Contenido Complejo de CI_Contact_Type (imagen generada con XMLSpy)

- 3) Todas las Clases UML que siguen este formato derivan de una clase Abstracta (*gco:AbstractObject_Type*) y extienden sus propiedades (atributos).
 Veamos el diagrama de la clase *gco:AbstractObject_Type*:

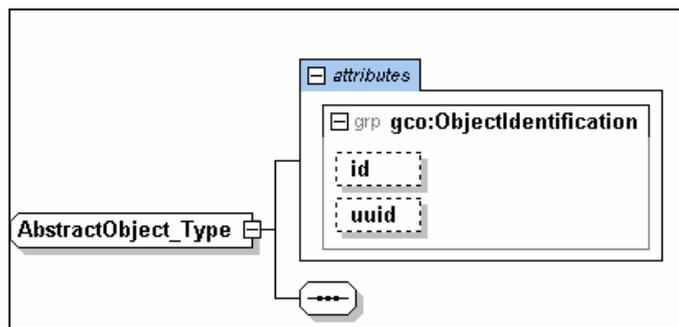


Figura 4. Diagrama de AbstractObject_Type (imagen generada con XMLSpy)

Esta herencia de los atributos *id* y *uuid* que proceden de *gco:AbstractObject_Type*, se hace para cumplir con la norma ISO19118 y su descripción de dos identificadores requeridos para identificar los objetos.

- 4) Por último la clase de Tipo Complejo tendrá una secuencia de elementos que identificarán a cada una de las propiedades de la clase. Una secuencia en el lenguaje de esquemas XML obliga a que todos los elementos sigan el orden con el que se definen en el esquema. La definición de un elemento en el lenguaje de esquemas utiliza varios atributos:
- *name*: el nombre del elemento.
 - *type*: tipo de dato.
 - *minOccurs*: número mínimo de apariciones de este elemento.
 - *maxOccurs*: número máximo de apariciones de este elemento.

El diagrama completo de la clase *CI_Contact* sería el siguiente:

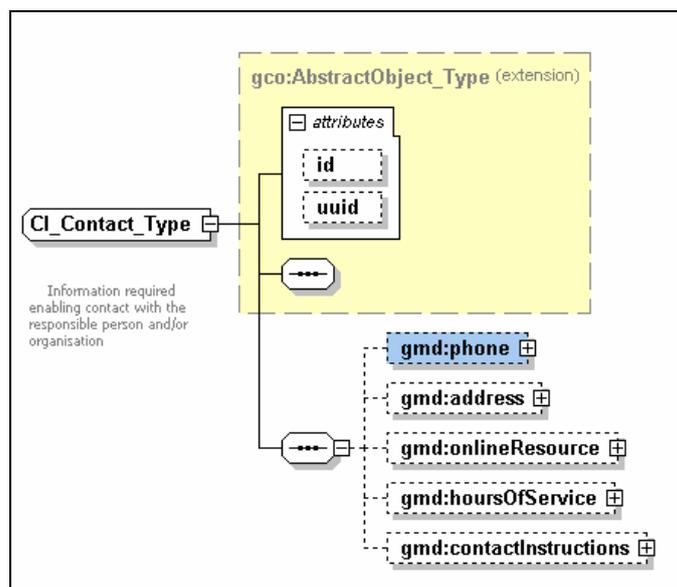


Figura 5. Diagrama de `CI_Contact_Type` (imagen generada con XMLSpy)

2.2.2 Definición de un Elemento Global

Además de definir un Tipo de Clase Compleja para todas las clases UML, se define un tipo de Elemento Global con el nombre que coincide con el nombre del elemento en la norma ISO19115 y cuyo tipo es el que se ha definido previamente:

```
<!-- ..... -->
<xs:element name="CI_Contact" type="gmd:CI_Contact_Type"/>
<!-- ..... -->
```

Figura 6. Definición del elemento `CI_Contact` (imagen generada con XMLSpy)

Este elemento global es identificado en la norma ISO19139 como un XCGE (XML Class Global Element).

2.2.3 Definición de un Tipo Propiedad Complejo

Ya se ha comentado que una propiedad de una clase UML puede ser un atributo, pero también una relación (asociación, agregación, composición) con otras clases. Para cubrir el requerimiento de que una clase sea una propiedad de este tipo dentro un contenedor de clases, se define un tipo de dato Complejo que en la norma ISO19139 se denomina como XCPT (XML Class Property Type) y que tiene el nombre del elemento de la norma ISO19115 seguido del sufijo: “`_PropertyType`”. Así, para `CI_Contact` se crea el Tipo de Propiedad Complejo `CI_Contact_PropertyType`.

Si se observa en la norma ISO19115, se puede comprobar que muchos elementos tienen como tipo de dato una asociación, una clase especificada o agregada. Esto también se puede ver claramente en la representación de la jerarquía de metadatos que muestra el software IME en la figura 7.

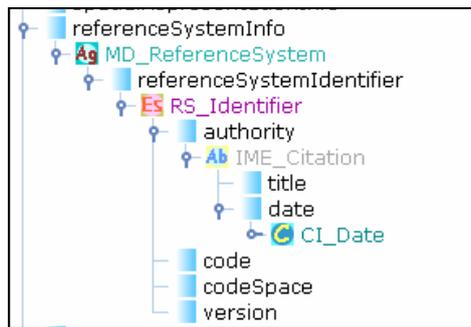


Figura 7. Representación de la entidad *MD_ReferenceSystem* en IME 4.0 en la que se pueden distinguir los tipos: Agregada, Especificada, Abstracta y Clase

Para incluir esta relación entre elementos, dentro del esquema XML, se recurre a este tipo de dato complejo XCPT (XML Class Property Type) formado por varias partes:

```
<xs:complexType name="CI_Contact_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="gmd:CI_Contact"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

Figura 8. XCPT para la entidad *CI_Contact* (imagen generada con XMLSpy)

En primer lugar una secuencia que contiene un único elemento que hace referencia al elemento global comentado anteriormente: *gmd:CI_Contact*.

Es decir, este tipo de dato puede contener directamente un elemento de tipo *gmd:CI_Contact* (asignación por valor).

Además se definen dos Atributos que permiten que este tipo de dato contenga, en lugar del valor de un elemento, una referencia a otro elemento:

- Un grupo del tipo *gco:ObjectReference*
- Un atributo del tipo *gco:nilReason*

El grupo *gco:ObjectReference* permite hacer referencia a otros objetos tal y como se define en la norma ISO19118.

Sin embargo, el modelo de contenido de *ObjectReference* definido en la ISO19118 ha sido ligeramente modificado para ser más consistente con la ISO19136 [9] y dar soporte a *XLink* (lenguaje definido en términos de marcas XML que permite introducir enlaces en los ficheros XML, de modo que podamos relacionar unos ficheros con otros).

Incluye también un atributo: *uuidref* que permite referenciar a otro objeto que se haya identificado mediante un atributo *uuid*. Un *uuid* es un Identificador Único Universal que se puede generar mediante varios métodos y que se crea a partir de un algoritmo que garantiza la unicidad del identificador en tiempo y espacio. No depende de ninguna autoridad de registro.

Al utilizar estos identificadores universales, se permite que el objeto referenciado se encuentre en el mismo Conjunto de Datos o en el mismo dominio de aplicación al que pertenezcan estos datos.

En el punto 3 se muestra un ejemplo de cómo se trabaja con este sistema por referencia a objetos.

Por último, se incluye el atributo *gco:nilReason* para permitir que una propiedad que se debe

representar como una referencia, tenga un contenido vacío. Los valores posibles de este atributo son: “*inaplicable*”, “*missing*”, “*template*”, “*unknown*”, “*withheld*”.

En la figura 9 se puede ver el diagrama de *CI_Contact_PropertyType*.

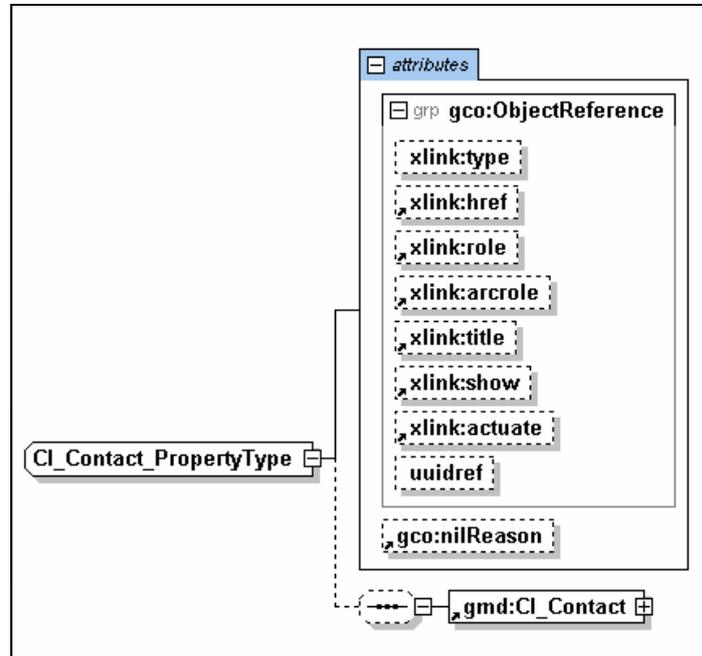


Figura 9. Diagrama de *CI_Contact_PropertyType* (XCPT de *CI_Contact*).
Imagen creada con XMLSpy.

Para entender el uso que se hace en el esquema de la ISO19139 de estos Tipos de Propiedad Complejos (XCPT), veamos la definición de otra entidad: *CI_ResponsibleParty*. Igual que se ha explicado para *CI_Contact*, tenemos tres declaraciones para *CI_ResponsibleParty*:

En la ISO19115, la entidad tiene el nombre *CI_ResponsibleParty*.

En el esquema se definen:

El Tipo Complejo que expande la clase *gco:AbstractObjectType* y añade las propiedades de *CI_ResponsibleParty*.

```
<xs:complexType name="CI_ResponsibleParty_Type">... </xs:complexType>
```

El Elemento Global que representa a la entidad de la ISO19115 *CI_ResponsibleParty*:

```
<xs:element name="CI_ResponsibleParty" type="gmd:CI_ResponsibleParty_Type"/>
```

El Tipo de Propiedad Complejo que se utilizará en las declaraciones de propiedades del tipo *CI_ResponsibleParty* que se tengan que incluir en otras entidades:

```
<xs:complexType name="CI_ResponsibleParty_PropertyType">... </xs:complexType>
```

En el Tipo Complejo *gmd:CI_ResponsibleParty_Type* se incluye un elemento *contactInfo* que según la ISO19115 es del Tipo Clase *CI_Citation*:

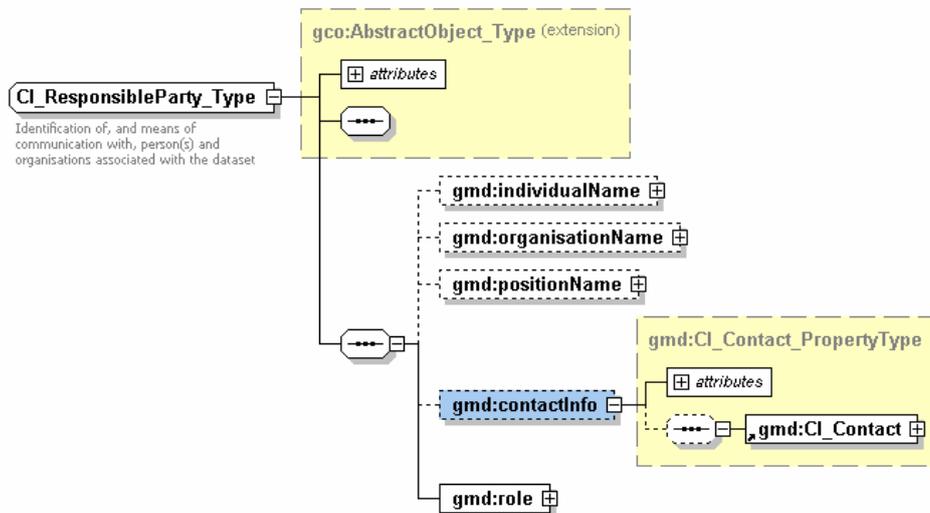


Figura 9. Diagrama de Tipo Complejo *CI_ResponsibleParty_Type*
(imagen generada con XMLSpy)

Se puede observar en la figura anterior que para indicar el tipo de este metadato (*gmd:contactInfo*) se utiliza el Tipo de Propiedad Complejo *gmd:CI_Contact_PropertyType*.

Al hacer esta declaración tenemos tres posibilidades para incluir valores en *gmd:contactInfo* en un teórico fichero XML basado en este esquema:

- Un elemento *CI_Contact* incluido directamente bajo *gmd:contactInfo*.
- Una referencia a otro *CI_Contact*.
- Una referencia a un valor nulo.

2.3 Definición de Tipos de Clases para Estereotipos

Los pasos descritos anteriormente se siguen, en todo el esquema, para la mayoría de las Clases UML de la ISO19115. Sin embargo hay varias excepciones basadas en los estereotipos que se aplican a algunas clases. Por norma, una clase sin estereotipo asignado o con el estereotipo Type o DataType seguirá la estructura que se acaba de explicar.

Otras clases que siguen patrones diferentes son:

- Clases Abstractas
- Herencia y subclases
- Enumeraciones
- Codelists
- Uniones

Por motivos de espacio, sus peculiaridades no serán tratadas en este artículo.

3 Validación de ficheros XML

Veamos el comportamiento de algunos casos de uso al generar documentos XML e intentar su validación con el esquema:

3.1 Metadatos obligatorios

El esquema no verifica que estén incluidos los metadatos obligatorios declarados en la norma ISO19115. En la definición del esquema no se ha incluido la verificación de la existencia de estos elementos en un fichero XML

Por ejemplo, el siguiente fichero XML valida correctamente con el esquema ISO19139, a pesar de que los elementos *contact*, *dateStamp* e *identificationInfo* están vacíos y no incluyen sus metadatos obligatorios.

```
<?xml version="1.0" encoding="UTF-8"?>
<MD_Metadata
  xmlns="http://www.isotc211.org/2005/gmd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.isotc211.org/2005/gmd
  C:\IME_V4_0\19139_ISO\gmdlmetadataEntity.xsd">
  <contact/>
  <dateStamp/>
  <identificationInfo/>
</MD_Metadata>
```

Figura 10. Diagrama de fichero XML validado con el esquema (imagen generada con XMLSpy)

3.2 Clase por valor o por referencia

El Tipo Propiedad Complejo que se ha explicado en el punto 2.2 demuestra que es posible resolver el valor de un objeto tanto por valor como por referencia. Es decir, puede incluirse un nuevo objeto de la estructura XML correspondiente, con valores para todos los metadatos (elementos) que se vayan a utilizar, pero también existe la posibilidad de hacer referencia a otro objeto del mismo tipo que se haya declarado o dentro del mismo fichero XML o dentro del dominio de aplicación.

En la figura 11 se observa como *contactInfo* de *CI_ResponsibleParty* se resuelve con un atributo *uuidref* que apunta al *uuid* de la entidad *CI_Contact* que se definió previamente dentro de otra objeto *CI_ResponsibleParty*:

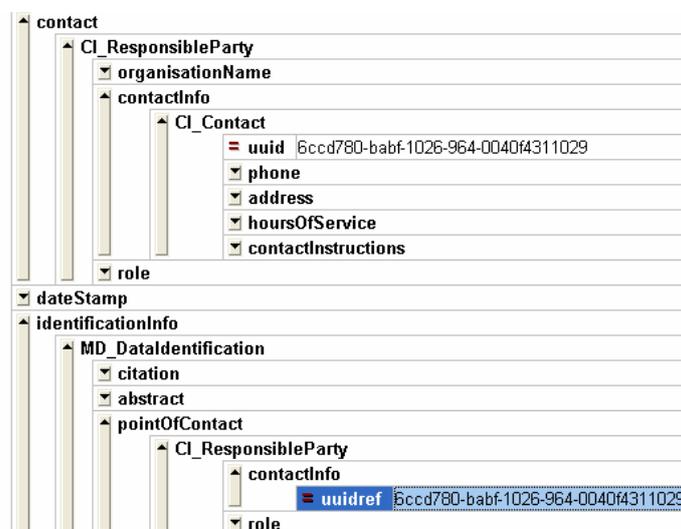


Figura 11. Diagrama de fichero asignación de una entidad por Referencia

(imagen generada con XMLSpy)

Este sistema amplía enormemente las posibilidades de creación de ficheros XML, permite reducir la duplicidad de información, pero añade complejidad a la validación de los ficheros XML y al proceso de los datos que deben realizar las aplicaciones que quieran procesar los metadatos.

3.3 Reglas de la ISO19115 que no han podido ser implementadas.

En la norma ISO19139 se hace referencia a algunas reglas especificadas en la norma ISO19115, en cuanto a la relación y valores posibles de los metadatos, que no han podido ser implementadas en el lenguaje de esquemas XML.

Por ejemplo, no es posible forzar a que una propiedad sólo pueda tener contenido “por valor” o “por referencia” o que incluya un atributo “nilReason” de nulidad. Haría falta recurrir a otros métodos de validación de XML (Schematron o transformaciones XSL).

Tampoco es posible obligar a que se incluya el metadato *units* en MD_Band si se están usando *maxValue* o *minValue*.

En el Anexo A de la norma ISO19139 es posible encontrar una tabla con todas las reglas que no se han podido forzar con el lenguaje del esquema. Es importante consultarlas porque será el creador de los ficheros XML el encargado de controlar estos puntos para mantener la interoperabilidad de sus datos.

3.4 Las Clases especificadas no pueden coincidir.

Según el esquema de la ISO19139, dos clases especificadas que tengan un mismo padre no pueden coincidir en el fichero XML porque se producirá un error de validación.

Esto lleva a problemas como el siguiente:



Figura 12. Metadato metadataConstraints (imagen generada con IME4.0)

En la figura anterior se puede apreciar como *metadataConstraints* tiene dos clases hijas del tipo especificadas: *MD_LegalConstraints* y *MD_SecurityConstraints*.

Si miramos cómo se ha definido en el esquema el elemento *metadataConstraints*:

```
<xs:element name="metadataConstraints" type="gmd:MD_Constraints_PropertyType"
minOccurs="0" maxOccurs="unbounded"/>
```

Y vemos la definición del Tipo Propiedad Complejo *gmd:MD_Constraints_PropertyType* de la figura 13.

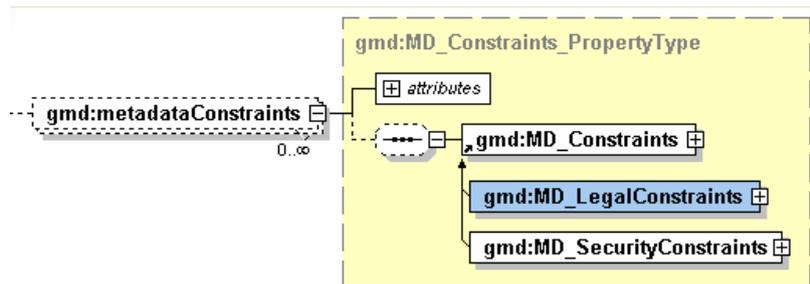


Figura 13. Diagrama de metadataConstraints (imagen generada con XMLSpy)

La clase `gmd:MD_Constraints` tiene dos clases que pertenecen a su Grupo de Sustitución: `gmd:LegalConstraints` y `gmd:MD_SecurityConstraints`. El concepto de Grupo de Sustitución existe en el lenguaje de esquemas para permitir que un elemento XML pueda ser sustituido por otro u otros manteniéndose la validez del documento.

Es decir, `gmd:MetadataConstraints` podría tener como elemento hijo o bien `gmd:LegalConstraints` o bien `gmd:MD_SecurityConstraints`. Según este criterio, para un Conjunto de Datos no se pueden especificar restricciones legales y de seguridad al mismo tiempo en el fichero XML y sería necesario duplicar el elemento `gmd:metadataConstraints` para incluir en uno las restricciones Legales y en otro las restricciones de Seguridad.

Los grupos de sustitución se utilizan mucho en el esquema de la ISO19139 y, a veces, pueden llevar a la validación de ficheros erróneos.

Así, todos los codelists se definen como pertenecientes al grupo de sustitución de `CharacterString`. Esto significa que si, por ejemplo, se crea un elemento `abstract` como el siguiente

```
<abstract>
  <MD_KeywordTypeCode codeListValue="" codeList=""/>
</abstract>
```

El fichero es válido según el esquema; a pesar de que realmente `abstract` debería ser utilizado así:

```
<abstract>
  <gco:CharacterString/>
</abstract>
```

4 Conclusiones

En este artículo se han explicado algunos matices y problemas que se empiezan a presentar al utilizar el esquema XML de la ISO19139, comentarios relacionados con la excesiva complejidad del esquema, a los que habría que añadir una frase que ya se ha citado en este artículo: “*Se podrían crear ficheros de metadatos en XML que serían válidos según el esquema, pero que no resultarían válidos desde el punto de vista de la norma ISO19115 y de la interoperabilidad*”.

Para reducir estos riesgos es necesario acotar, por medio de recomendaciones, el amplio campo de opciones que ofrece el esquema a la hora de validar ficheros XML. A la espera de que se inicie un uso masivo de la norma ISO19139 y se incrementen los comentarios de los grupos de expertos y de los organismos encargados de verificar la interoperabilidad, el margen de documentos XML que resultan correctos pero no intercambiables es demasiado elevado como para que las aplicaciones de

proceso y catálogo de metadatos puedan empezar a concretar soluciones para los usuarios.

En este artículo se ha hecho referencia a la aplicación IME del INTA. Un último comentario relativo a esta aplicación en relación con su visor del esquema. El principal problema que surge al intentar crear documentos XML es conocer cómo debe ser la estructura XML para que sean válidos. Esto hizo que para el desarrollo del visor del esquema se trabajara de un modo inverso, es decir, la aplicación carga el esquema de la ISO19139 completo y a partir de este modelo se construyen las estructuras en XML de cada entidad o elemento de la ISO19115.

En este artículo ya se ha comentado que siempre existirán varias soluciones correctas, especialmente en el tema de la inclusión de objetos por valor o por referencia, pero en la versión 4.0 de IME se puede visualizar, con las limitaciones del método recursivo empleado, al menos una de ellas.

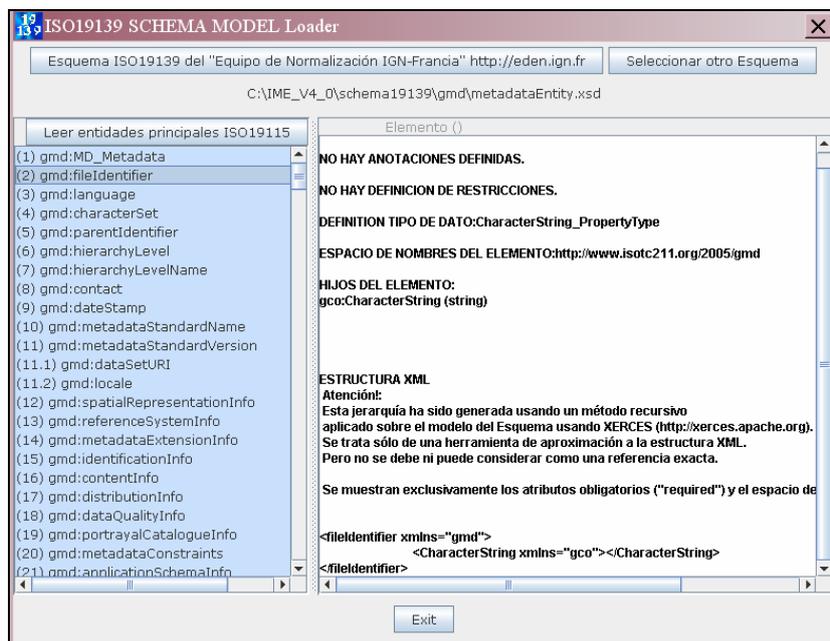


Figura 14. Visor del Esquema de IME 4.0

Referencias

- [1] ISO 19115:2003 Geographic information - Metadata
- [2] E.D.E.N Equipo de expertos en Normalización del IGN-Francia (<http://eden.ign.fr>)
- [3] Repositorio de Esquemas de la ISO (<http://standards.iso.org/ittf/PubliclyAvailableStandards>)
- [4] ISO 19139:2007 Geographic information - Metadata - XML schema implementation.
- [5] IME (ISO Metadata Editor del INTA) (<http://www.crepad.rcanaria.es/metadata/index.htm>)
- [6] ISO 19103:2005 Geographic information - Conceptual schema language.
- [7] ISO 19118:2005 Geographic information - Encoding.
- [8] GML Geographic Markup Language
- [9] ISO/PRF 19136 Geographic information - Geography Markup Language (GML)