



# MEDIN Metadata Profile

## Schematron Validation

Version 1.0

2009-11-06



Better Data, Better Science, Improved Decisions

SeaZone provides Marine Geographic Information Solutions from instrument to desktop, supporting decision making in the Marine Environment and Coastal Zone

We combine our knowledge of marine science, data acquisition and use, with expertise in geographic information systems (GIS) and data management, to provide customers with innovative solutions that address their data access, processing and management needs.

SeaZone Solutions Limited,  
Red Lion House,  
Bentley,  
GU10 5HY

T: +44 (0) 870 013 0607  
F: +44 (0) 870 013 0608

E: info {at} seazone {dot} com

W: <http://www.seazone.com>

SeaZone Report Template 2.3

PROJECT NUMBER:			DOCUMENT REF:		
Version 1.0 - Release	JR				2009-11-06
Version 0.1	JR				2009-04-09
	Originated	Checked	Reviewed	Authorised	Date
Purpose / Description					

### Copyright

The Schematron schema referred to in this document and any associated material may be used and reproduced free of charge provided that it is done so accurately and not in any misleading context or in a derogatory manner. The Schematron schema must be acknowledged as being sourced from the Marine Environmental Information Network (MEDIN) and SeaZone Solutions Limited (SeaZone) and this cited when it is reproduced as part of another schema, software, publication or service.

### Disclaimer

The Schematron schema referred to in this document and any associated material has been compiled from sources believed to be proven in practice and reliable but no warranty, expressed or implied, is given that it is complete or accurate nor that it is fit for any particular purpose. All such warranties are expressly disclaimed and excluded and users are therefore recommended to seek professional advice prior to its use.

## CONTENTS

<b>1. INTRODUCTION</b>	<b>1</b>
<b>2. ABBREVIATIONS</b>	<b>1</b>
<b>3. ISO SCHEMATRON</b>	<b>2</b>
<b>4. SCHEMATRON VALIDATION</b>	<b>2</b>
4.1 Validation Mechanism	2
4.2 Patterns	3
4.3 Abstract Patterns	3
<b>5. METADATA VALIDATION PROCESS</b>	<b>3</b>
<b>6. DELIVERABLES</b>	<b>4</b>
<b>ANNEX A SCHEMATRON EXAMPLES</b>	<b>5</b>
A.1 Schematron Schema	5
A.2 Schematron Validation Report Language	6

### List of Tables

Table 1 – Abbreviations	1
Table 2 – Deliverables	4

### List of Figures

Figure 1 – Schematron Stylesheet Transformation Steps	2
Figure 2 – Schematron Pattern	3
Figure 3 – Metadata Validation Workflow	4

## 1. INTRODUCTION

ISO 19115 defines a set of elements for recording metadata and sets out a minimal set of constraints which must be achieved by all compliant profiles. A profile is a domain specific metadata set which can apply further constraints to ISO 19115 to achieve particular requirements. ISO 19139 is a technical specification which defines a set of schemas in XML Schema Definition language which set out an XML grammar for encoding ISO 19115 metadata in XML. XML can be assessed against schemas to ensure that the structure of the XML conforms to the structure defined in the schema. When the structure does conform the XML is said to be schema valid. A general limitation of XML schemas is that they are grammar based which means that they do not provide a means of validating element values or domain specific profiles of ISO 19115.

Schematron provides another way of validating XML by looking for tree patterns and element content. Schematron works by making assertions about the XML which resolve to true or false. If an assertion resolves to false it fails and the overall validation fails. The assertions are written using XPath in a Schematron schema (\*.sch) which is itself expressed in XML. Schematron is designed to be used in conjunction with other validation processes and is based on XSLT and XPath so it is simple to implement.

Indeed, the ISO / TS 19139 standard refers to Schematron as a means of testing some constraints that are set by ISO 19115 but are not testable with XSD. These constraints are set out in Table A.1 of ISO / TS 19139.

Schematron has been standardised by ISO as ISO 19757 Part 3. It is this version of Schematron that has been used to create the MEDIN Schematron schema.

The home of Schematron is <http://www.schematron.com/>.

This report documents the Schematron schema for the validation of XML metadata sets encoded in ISO 19139 XML according to the MEDIN Metadata Profile.

## 2. ABBREVIATIONS

**Table 1 – Abbreviations**

ISO	International Organisation for Standardisation
MEDIN	Marine Environmental Data and Information Network
SVRL	Schematron Validation Report Language
XML	eXtensible Mark-up Language
XPath	XML Path Language
XSD	XML Schema Definition
XSL	eXtensible Stylesheet Language
XSLT	XSL Transformation

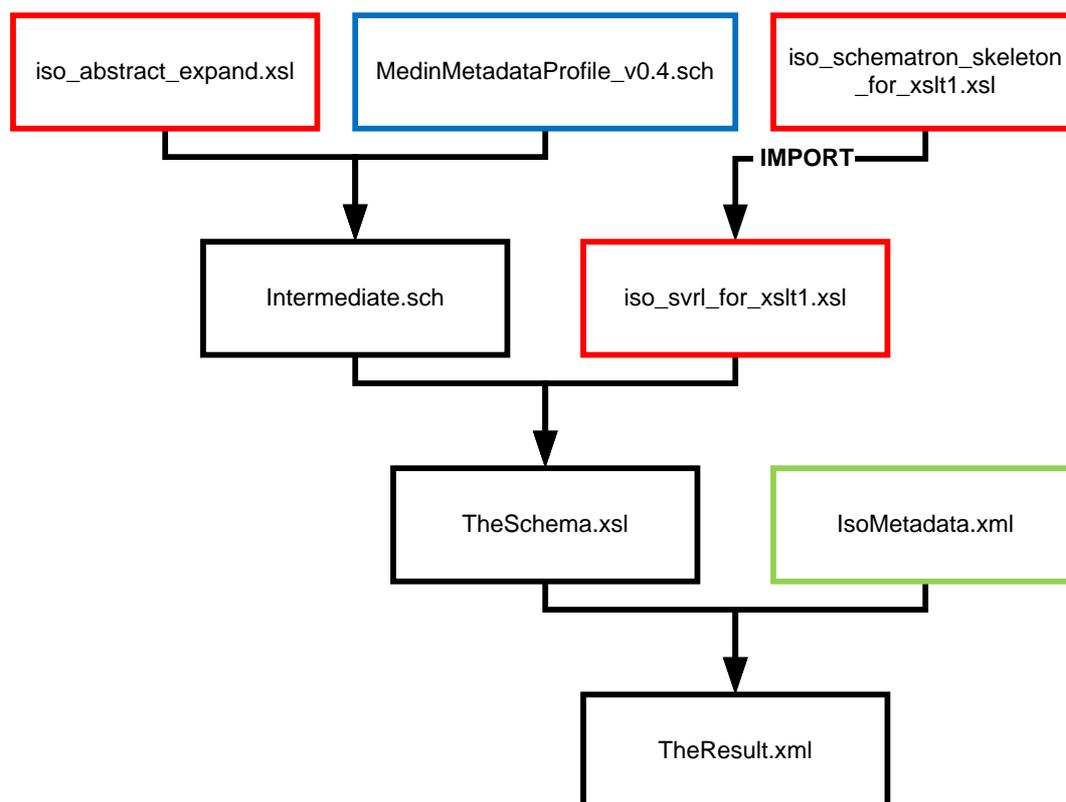
### 3. ISO SCHEMATRON

Schematron has been standardised by ISO as part of ISO 19757.<sup>1</sup> This standard is one of a set that are publicly available.<sup>2</sup> The ISO implementation of Schematron has been used to produce the MEDIN Schematron schema.

## 4. SCHEMATRON VALIDATION

### 4.1 Validation Mechanism

Mechanically, Schematron validation is implemented as a concatenated series of XSL Stylesheet transformations. The process is shown in Figure 1 where the red boxes represent the ISO Schematron stylesheets.<sup>3</sup>



**Figure 1 – Schematron Stylesheet Transformation Steps**

The blue box represents a Schematron schema which is developed for a domain specific purpose, in this case for validating the MEDIN metadata profile. An example of a Schematron schema can be seen in A.1. The Schematron schema is first styled by the `iso_abstract_expand.xsl` stylesheet to produce an intermediate Schematron schema. This step is only necessary if the initial schema contains abstract patterns, which is the case for the MEDIN schema. The intermediate schema is then styled with the `iso_svrl_for_xslt1.xsl` stylesheet. This produces another stylesheet which is used to style (validate) a candidate XML file which is shown as the green box. In our case the candidate XML file is a schema valid ISO 19139 encoded metadata set. The result of this transformation is a result set which is encoded in SVRL. An example of an SVRL output file is presented in A.2.

<sup>1</sup> ISO/IEC 19757-3:2006(E) Information Technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation - Schematron

<sup>2</sup> <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html> (search for Schematron)

<sup>3</sup> <http://www.schematron.com/tmp/iso-schematron-xslt1.zip>

## 4.2 Patterns

A pattern in Schematron is a high level structure for encapsulating a set of rules that are in some way related. Rules contain either assertions or reports both of which express tests written in XPath. An example of a pattern is shown in Figure 2. This pattern includes a title and a single rule. The rule contains two assertions and two reports which work in conjunction with each other so that if an assertion passes (returns true) the report also passes. In this case the value of the report gets written to the SVRL file so that it is obvious that a test has fired and passed. If the assertion fails its value is written to the SVRL file. This approach has been adopted in the schema: assertions and reports are typically implemented in pairs. Failed assertions imply non-conformity with the profile. Reports are included for information.

This pattern first tests that one element called `gmd:hierarchyLevel` is encoded in the context of `gmd:MD_Metadata`. The element `hierarchyLevel` is used to encode the Resource Type metadata. In the native ISO 19115 standard `gmd:hierarchyLevel` is an optional element and more than one may be encoded – its cardinality is `0..*`. However, in the MEDIN metadata profile this cardinality is changed to `1` – ie it becomes mandatory. The first assertion enforces this constraint.

The second assertion ensures that the value of `hierarchyLevel` is one of `dataset`, `series` or `service`. Again, in the native ISO 19115 standard the list of possible values is longer but the MEDIN metadata standard restricts it.

```
<sch:pattern fpi="MedinResourceType">
  <sch:title>Element 4 - Resource Type (M)</sch:title>
  <sch:rule context="/gmd:MD_Metadata">
    <sch:assert test="count(gmd:hierarchyLevel) = 1">
      Resource Type is mandatory. One occurrence is allowed.
    </sch:assert>
    <sch:report test="count(gmd:hierarchyLevel) = 1">
      Resource Type test passed.
    </sch:report>
    <sch:assert test="gmd:hierarchyLevel/gmd:MD_ScopeCode = 'dataset' or
      gmd:hierarchyLevel/gmd:MD_ScopeCode = 'series' or
      gmd:hierarchyLevel/gmd:MD_ScopeCode = 'service' ">
      Value of Resource Type must be dataset, series or service.
      Value of Resource Type is
      '<sch:value-of select="gmd:hierarchyLevel/gmd:MD_ScopeCode"/>'
    </sch:assert>"
    <sch:report test="gmd:hierarchyLevel/gmd:MD_ScopeCode = 'dataset' or
      gmd:hierarchyLevel/gmd:MD_ScopeCode = 'series' or
      gmd:hierarchyLevel/gmd:MD_ScopeCode = 'service' ">
      Value of Resource Type is
      '<sch:value-of select="gmd:hierarchyLevel/gmd:MD_ScopeCode"/>'
    </sch:report>
  </sch:rule>
</sch:pattern>
```

Figure 2 – Schematron Pattern

## 4.3 Abstract Patterns

Abstract patterns allow the definition of patterns which can be used to test structures which use different tree structures but which are fundamentally identical. Examples of common structures in ISO 19115 metadata are the responsible party data types which are used for encoding metadata, data and organisation contact information. Abstract patterns have been implemented in the MEDIN Schematron schema for this reason.

## 5. METADATA VALIDATION PROCESS

With the introduction of the Schematron schema to do the validation of the profile, the overall metadata validation workflow becomes a three stage process as shown in Figure 3. First a

candidate metadata set must be validated against the ISO 19139 schemas. If the metadata set proves to be schema valid it can then be validated against the ISO 19139 Table A.1 Constraints Schematron schema. The Schematron schema relies on hardcoded XPath statements which will only work effectively on a schema valid XML set. Finally, if the XML is still valid it can be validated against the MEDIN Profile Schematron Schema.

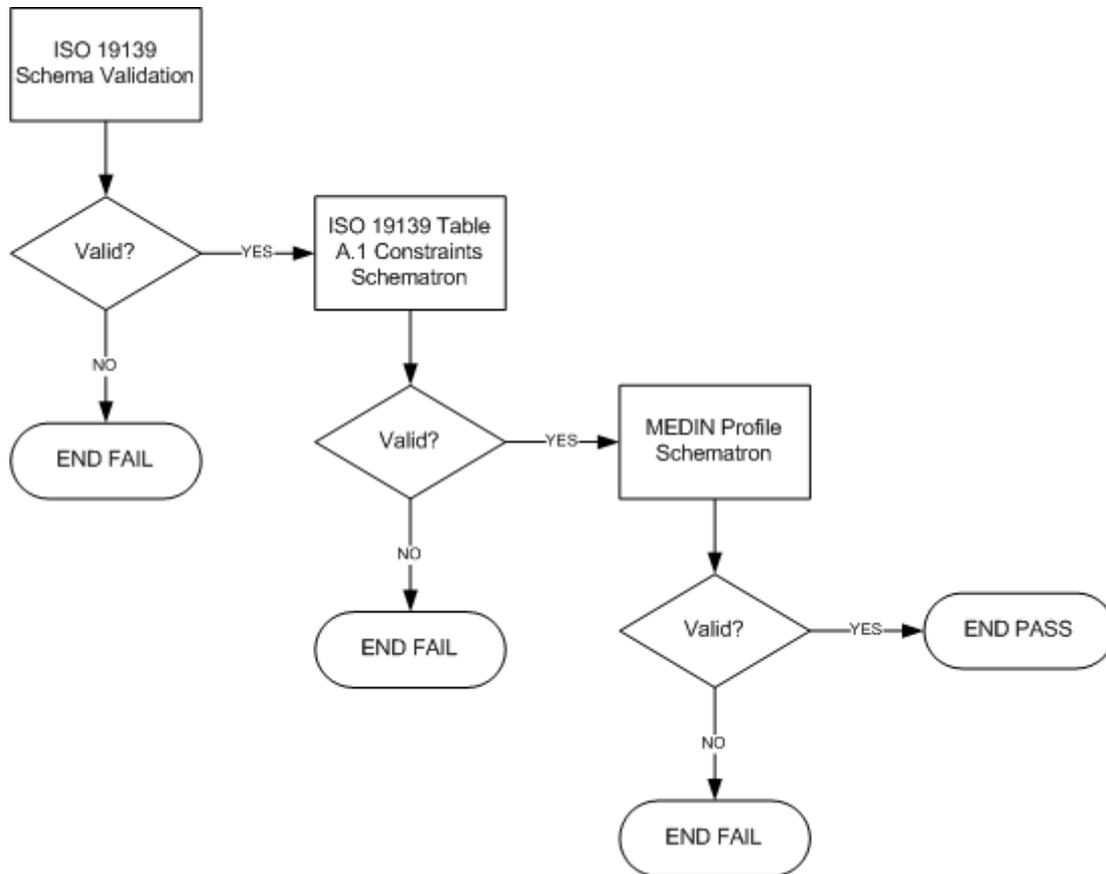


Figure 3 – Metadata Validation Workflow

## 6. DELIVERABLES

The items described in Table 2 are delivered as part of the output from this project.

Table 2 – Deliverables

Name	Description
ISOTS19139A1Constraints_v1.0.sch	Version 1.0 of the Schematron Schema for testing constraints set out in Table A.1 of ISO / TS 19139.
MedinMetadataProfile_v1.0.sch	Version 1.0 of the Schematron Schema for testing constraints set out in the MEDIN Metadata profile Version 2.3.

## Annex A Schematron Examples

### A.1 Schematron Schema

An example of a Schematron schema is presented below. It contains two concrete patterns, one set to produce successful output and one known to produce unsuccessful output, given a particular XML instance to validate. The patterns include assert elements and report elements which include test attributes and text which will be output to the SVRL in the event that the test returns true. Note that it is possible to include references which are resolved at runtime in the text using elements such as value-of.

An abstract pattern is also included in the example. Abstract patterns are not dissimilar to concrete patterns. The key differences are that they have an abstract attribute which must be set to true and they take parameters which are identified by the dollar sign. Concrete patterns implement the abstract pattern and when this is done the concrete pattern must include an is-a attribute which identifies its base abstract pattern. The param elements set the value of the abstract pattern parameters.

```
<?xml version="1.0" encoding="utf-8" ?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
queryBinding="xslt">

  <!-- Namespaces from ISO 19139 Metadata encoding -->
  <sch:ns prefix="gml" uri="http://www.opengis.net/gml" />
  <sch:ns prefix="gmd" uri="http://www.isotc211.org/2005/gmd"/>
  <sch:ns prefix="gco" uri="http://www.isotc211.org/2005/gco"/>
  <sch:ns prefix="gmx" uri="http://www.isotc211.org/2005/gmx"/>
  <sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>

  <!-- ===== -->
  <!-- Concrete Pattern - Set up to produce successful reports -->
  <!-- ===== -->

  <sch:pattern fpi="ExampleConcretePatternPass">
    <sch:title>Example Pattern - Pass</sch:title>
    <sch:rule context="//gmd:CI_ResponsibleParty">
      <sch:assert test="count(gmd:organisationName) +
        count(gmd:individualName) &gt;= 1">
        At least one organisation name and / or individual
        name must be provided.
      </sch:assert>
      <sch:report test="count(gmd:organisationName) = 1">
        The value of organisationName is
        '<sch:value-of select="gmd:organisationName"/>'.
      </sch:report>
      <sch:report test="count(gmd:individualName) = 1">
        The value of individualName is
        '<sch:value-of select="gmd:individualName"/>'.
      </sch:report>
    </sch:rule>
  </sch:pattern>

  <!-- ===== -->
  <!-- Concrete Pattern - Set up to produce failed asserts -->
  <!-- ===== -->

  <sch:pattern fpi="ExampleConcretePatternFail">
    <sch:title>Example Pattern - Fail</sch:title>
    <sch:rule context="//gmd:CI_ResponsibleParty">
```

```

    <sch:assert test="count(gmd:organisationName) +
        count(gmd:individualName) >= 10">
        At least 10 organisation names and / or individual
        names must be provided. Only
        <sch:value-of select="count(gmd:organisationName) +
            count(gmd:individualName)"/> found.
    </sch:assert>
</sch:rule>
</sch:pattern>

<!-- ===== -->
<!-- Concrete pattern implementing abstract pattern -->
<!-- ===== -->

<sch:pattern is-a="GcoTypeTestPattern" id="ExampleGcoTypeTest">
  <sch:title>Example Pattern - Call Abstract Pattern</sch:title>
  <sch:param name="context"
    value="//gmd:identificationInfo/*/gmd:citation/*/gmd:title"/>
  <sch:param name="element"
    value="gco:CharacterString"/>
</sch:pattern>

<!-- ===== -->
<!-- Abstract Pattern -->
<!-- ===== -->

<sch:pattern abstract="true" id="GcoTypeTestPattern">
  <sch:rule context="$context">
    <sch:assert test="(string-length($element) > 0) or
      (@gco:nilReason = 'inapplicable' or
      @gco:nilReason = 'missing' or
      @gco:nilReason = 'template' or
      @gco:nilReason = 'unknown' or
      @gco:nilReason = 'withheld')">
      The <sch:name/> element must have a value or a Nil Reason.
    </sch:assert>
    <sch:report test="(string-length($element) > 0)">
      The <sch:name/> element has a value of
      '<sch:value-of select="$element"/>'.
    </sch:report>
    <sch:report test="(@gco:nilReason = 'inapplicable' or
      @gco:nilReason = 'missing' or
      @gco:nilReason = 'template' or
      @gco:nilReason = 'unknown' or
      @gco:nilReason = 'withheld')">
      The <sch:name/> element has a Nil Reason attribute with a value of
      '<sch:value-of select="@gco:nilReason"/>'.
    </sch:report>
  </sch:rule>
</sch:pattern>
</sch:schema>

```

## A.2 Schematron Validation Report Language

The following is the output from the validation of an ISO 19139 encoded XML file using the Schematron schema shown in A.1. Note that it is incomplete and that some report elements have been replaced with an ellipsis to make it more easily readable. The data indicates that the pattern entitled “Example Pattern – Pass” has fired (see bold element active-pattern). The bold successful-report element indicates that the report with test `count(gmd:organisationName) = 1` has returned successfully. Its text element indicates the output and that the value of the

gmd:organisationName that was found is 'SeaZone Solutions Limited'. The full version of the file then contains a series of successful-report elements.

When the active pattern becomes "Example Pattern – Fail", tests begin to fail. This is identified from the presence of the failed-assert element. The test here is that the count of gmd:organisationName and gmd:individualName for a particular gmd:CI\_ResponsibleParty element must be at least 10 (the &gt; escape sequence is interpreted in XML as >). The text element reports the failure and also the total number of the respective gmd elements that were found.

```
<?xml version="1.0" encoding="utf-16" standalone="yes"?>
<svrl:schematron-output title="" schemaVersion=""
  xmlns:svrl="http://purl.oclc.org/dsdl/svrl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:schold="http://www.ascc.net/xml/schematron"
  xmlns:sch="http://www.ascc.net/xml/schematron"
  xmlns:iso="http://purl.oclc.org/dsdl/schematron"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:gmd="http://www.isotc211.org/2005/gmd"
  xmlns:gco="http://www.isotc211.org/2005/gco"
  xmlns:gmx="http://www.isotc211.org/2005/gmx"
  xmlns:xlink="http://www.w3.org/1999/xlink">

  <svrl:ns-prefix-in-attribute-values
    uri="http://www.opengis.net/gml" prefix="gml" />
<svrl:ns-prefix-in-attribute-values
  uri="http://www.isotc211.org/2005/gmd" prefix="gmd" />
  <svrl:ns-prefix-in-attribute-values
    uri="http://www.isotc211.org/2005/gco" prefix="gco" />
  <svrl:ns-prefix-in-attribute-values
    uri="http://www.isotc211.org/2005/gmx" prefix="gmx" />
  <svrl:ns-prefix-in-attribute-values
    uri="http://www.w3.org/1999/xlink" prefix="xlink" />

  <svrl:active-pattern name="Example Pattern - Pass" />
  <svrl:fi-red-rule context="//gmd:CI_ResponsibleParty" />
  <svrl:successful-report test="count(gmd:organisationName) = 1"
    location="/*[local-name()='MD_Metadata' and
namespace-uri()='http://www.isotc211.org/2005/gmd']/*[local-
name()='contact' and namespace-
uri()='http://www.isotc211.org/2005/gmd']/*[local-
name()='CI_ResponsibleParty' and namespace-
uri()='http://www.isotc211.org/2005/gmd']">
    <svrl:text>
      The value of organisationName is 'SeaZone Solutions Limited'.
    </svrl:text>
  </svrl:successful-report>
  ...
  <svrl:active-pattern name="Example Pattern - Fail" />
  <svrl:fi-red-rule context="//gmd:CI_ResponsibleParty" />
  <svrl:failed-assert test="count(gmd:organisationName) +
count(gmd:individualName) &gt;= 10"
    location="...">
    <svrl:text>
      At least 10 organisation names and / or individual
      names must be provided. Only 2 found.
    </svrl:text>
  </svrl:failed-assert>
  ...
  <svrl:active-pattern id="ExampleGcoTypeTest" name="ExampleGcoTypeTest"
/>
```

```
<svrl:-fired-rule
context="//gmd:identificationInfo/*/gmd:citation/*/gmd:title" />
<svrl:successful-report test="(string-length(gco:CharacterString) >
0)"
                        location="...">
  <svrl:text>
    The gmd:title element has a value of 'A Dataset Title'.
  </svrl:text>
</svrl:successful-report>
</svrl:schematron-output>
```