ISO/TC 211 N 419

1997-10-01

Number of pages: 4

ISO/TC 211
Geographic information/Geomatics

Title: Guidelines on Abstract Test Suites and Conformance Clauses

Source: WG 1, Mr. Kenichi Asai

Required Action: For information and consideration at the 5th plenary meeting in Oxford, UK, 1997-10-02/03

File name(s): 211n419.doc, 211n419.PDF

Distribution: P, O and L members
Chairman
WG Convenors
Mr. Keith Brannon, ISO CS
Mr. François Salgé, CEN/TC 287

ISO/TC 211 Secretariat

Norwegian Technology Standards Institution
Oscars gate 20
P.O. Box 7072 Majorstua
N-0306 Oslo, Norway

Telephone: + 47 22 59 67 16
Telefax: + 47 22 59 67 33

E-mail: Bjornhild.Saeteroy@nts.no
or: bjs@nts.no
URL: http://www.statkart.no/isotc211/

Guidelines on Abstract Test Suite and Conformance Clause

For a standard to be effective and useful, it is essential that the conformance to the standard can be determined clearly. For this purpose, all project teams are responsible to write a clear, unambiguous, and testable standard. To help writing such a standard, Conformance and Testing project team provides here a general guideline and a checklist for each (testable) working item.

This document is divided into three parts. The first part describes how to structure the abstract test suite (ATS) of each part of the standard in a clear, hierarchical way. The second part describes how to write a conformance clause, which we require all the (testable) working items to include at the clause 2 of their documents. The third part shows a checklist to be fulfilled in each document.

Below, we will illustrate with an example how an ATS is structured, how to write a conformance clause, as well as a checklist to be satisfied in all (testable) working items. Here, we will use Quality working item as an example. Note that the example is just for illustration, and thus is impossible to use it as an actual document for Quality.

1 How to structure ATS

Since ATS is represented as a hierarchical structure of test purposes, the essential part of constructing ATS is the identification of test purposes.

1.1. Identifying test purpose

The first thing to do is to identify the main test purpose of each working item. This will become the test purpose of the root node of ATS. To be more specific, you have to answer the following question:

> "What does your part of the standard specify?"

The answer to this question usually becomes quite obvious. For example, it is "How to specify quality" for Quality working item. Thus, the test purpose of the root node of ATS becomes: "check the conformance to the definition of Quality".

1.2. Decomposing test purpose

Once the main test purpose is determined, we continue by decomposing it. The result of decomposition becomes the child nodes in ATS. To be more specific, you will answer the following question:

> "For an implementation to be in conformance to the test purpose specified, what requirements have to be satisfied?"

In our Quality example, it might be "It has to contain data quality elements and data quality overview elements." Thus, we have now two child nodes: "check the conformance of data quality elements" and "check the conformance of data quality overview elements".

1.3. Repeating identification and decomposition of test purposes

This process will continue until all test purposes are decomposed into sufficiently small pieces, i.e., abstract test cases. In our Quality example, data quality elements will be further decomposed into data quality subelements, and then into data quality measures, which would be basic enough to be abstract test cases. (The judgement whether it is basic enough or not should be done in each working item.)

On the other hand, data quality overview elements will be decomposed into three components: purpose, lineage, and usage, which would be all small enough to be abstract test cases.

1.4. Making the hierarchical structure

When all test purposes are decomposed into abstract test cases, all the test purposes are collected to form the ATS of the standard. In our Quality example, the final ATS would become as follows. Observe that the hierarchical structure of the test purposes automatically constitutes the structure of ATS.

2.1 ATS of Quality

- Test purpose:

check the conformance to the definition of Quality

- Test method:

check if it has two components, data quality elements (Clause 2.1.1) and data quality overview elements (Clause 2.1.2), and see both satisfy their requirements

2.1.1 Data quality elements

- Test purpose:

check the conformance to the definition of data quality elements

- Test method:

check if it has data quality subelements. If it has, check their conformance to the definition of data quality elements (Clause 2.1.1). Otherwise, for each data quality measures, check if they are actually a valid data quality measures as described in clause *.*.

2.1.2 Data quality overview elements

- Test purpose:

check the conformance to the definition of data quality overview elements

- Test method:

check if it has three components, purpose, lineage, and usage, and see if they are valid as described in clause *.*.

1.5. Notes on Test Methods

A typical test method for abstract test modules would be a reference to other clauses. It might also include statements on whether it is mandatory, optional, or conditional. As for abstract test cases, specific test methods have to be stated as well as their types: either a basic test or a capability test. Here, a basic test is a simple test that can be tested by each implementor of GIS systems, while a capability test is a more complex test that should be performed in a testing laboratory.

2. How to write a conformance clause

A conformance clause forms an entry point for testing conformance. When one wants to check if his implementation conforms to the standard, he first see the conformance clause to determine what he has to check. Thus, conformance clauses have to state clearly what requirements should be fulfilled to claim the conformance to the standard.

2.1 Basic conformance clause

If the standard is formed in a hierarchical way as described in section 1, a conformance clause becomes quite obvious. It will just say that the root requirement of ATS has to be fulfilled. In our Quality example, it will look like this:

> 2. Conformance (Clause for Quality)
> To check the conformance of a quality description to this part of standard, one has to satisfy all the requirements presented in Clause 2.1.

Note: Here, we assumed that the conformance clause includes ATS in Clause 2.1 (as in Section 1.4). In case ATS is omitted because the main body of the standard is well-structured as ATS, it should refer to the main body instead.

2.2. Extended conformance clause

The hierarchical structure of ATS is not always clear. For example, if the standard allows several alternatives, the root requirement of ATS would be one of those alternatives. It might also have some condition to choose the alternatives. In these cases, the conformance clause should designate all the root requirements (together with the condition, if any) explicitly. Such an example is:

> To claim conformance to this part of standard, one has to satisfy one of the requirements described in Clause *, Clause *, or Clause *.

2.3. Conformance class/level

In some cases, conformance classes are used to define different kinds of conformance requirements. If some parts of the standard are essential for some particular application but not the others, it is not preferable to specify them mandatory requirements, because it will then require all the systems to fulfill the requirements. To avoid this, conformance classes are used.

When conformance classes are used in the standard, they have to be defined in the conformance clause. For example,

> In this part of the standard, three classes of conformance are defined. To claim the conformance to the conformance class A, B, or C, one has to satisfy the requirements described in Clause *, *, or *, respectively.

The requirements of each conformance class can be overlapped. Added to the requirements fulfilled by each conformance class, its intended application should be also mentioned.

A conformance level is a special kind of conformance classes where requirements of a higher level includes all the requirements of lower levels. Below shows an example of defining three conformance levels in the conformance clause.

> To claim conformance to this part of standard, one has to satisfy one of three conformance levels defined below. The requirements that have to be satisfied are as follows:
> - conformance level 1 (low level)
>
> Clause * and Clause *.
> - conformance level 2 (medium level)
>
> Clause * in addition to the conformance level 1.
> - conformance level 3 (high level)
>
> Clause * in addition to the conformance level 2.

When conformance classes or levels are defined, conformance testing will be performed with respect to those specific conformance classes or levels.

2.4. Reference to main clauses

In conformance clauses, it is required to refer to the main clauses of the standard explicitly to clarify the entry points of conformance testing. In those cases where several entry points exist, they all have to be stated explicitly.

3. Checklist for Conformance and Testing

Once a conformance clause is clearly stated, there are almost nothing left to be done. Here, Conformance and Testing project team provides a checklist for quality control, which can be applied in each working item fairly easily after the construction of conformance clause.

- Does a conformance clause exist?
- Does it clearly state the requirements that have to be satisfied to claim the conformance to this part of the standard?
- Does it explicitly refer to the main clauses that become an entry point of conformance testing?
- Are conformance classes/levels defined in the conformance clause? (Applicable only when conformance classes/levels are used.)

3.1. When to check this checklist and who

We encourage each working item to check this checklist whenever each working item makes a new draft. We are also happy to check if the conformance clause written is appropriate.